FOREST DECISION MAKING UNDER UNCERTAINTY: ADAPTIVE MANAGEMENT FOR THE CONSERVATION OF BIRD POPULATIONS ON A NATIONAL WILDLIFE REFUGE

by

CLINTON THOMAS MOORE

(Under the direction of Michael J. Conroy)

ABSTRACT

I constructed a stochastic, spatially-explicit landscape model to seek optimal forest management decisions for long-term persistence of populations of red-cockaded woodpecker (*Picoides borealis*) and wood thrush (*Hylocichla mustelina*) on the Piedmont National Wildlife Refuge in Georgia, USA. I addressed uncertainty in decision making by considering alternative model forms that expressed different mechanisms of response by the forest and the bird populations to silvicultural actions. The implication of model uncertainty in this system is that conservation tradeoffs for both species differ according to choice of model. Decision variables in each model were the spatial scheduling of forest compartments for silvicultural treatments and the average periodicity of prescribed burning in compartments. Model responses were the number of active woodpecker clusters and abundance of wood thrushes. Additionally, I obtained a composite response as the average of the two abundance responses, each scaled by its standard error. I simulated each model under extremes of the decision alternatives, and I found a near-optimal management schedule for each model and for each of the responses. I also found near-optimal schedules for the case of complete uncertainty with regard to all models in the model set. Forest and bird monitoring data collected on the Refuge are the means by which measures of belief in each model are updated and decisions are adaptively improved. In nearly all models, both species responded strongly, but in opposite directions, to burning, and woodpeckers were sensitive to compartment scheduling. Consequently, optimal decisions were mostly similar among models, and values of information computed for each response suggested that little would be gained in management performance by resolving uncertainty among these models. However, fundamental uncertainties in the management of this system were probably not captured in this model set, and adaptive approaches therefore still hold promise for Refuge management. Current impediments to conducting adaptive management on the Refuge are (1) uncertainties regarding objectives, (2) lack of a comprehensive forest monitoring system, (3) inadequate system models, and (4) constraints in the expression and breadth of decision alternatives. I discuss critical information needed for the adaptive management of this and similar resource systems.

INDEX WORDS: Adaptive management, Birds, Conservation, Decision making, Dynamic model, Endangered species, Forest management, Georgia, Habitat model, *Hylocichla mustelina*, Monitoring, Multiple species management, Optimization, *Picoides borealis*, Piedmont National Wildlife Refuge, Population model, Red-cockaded woodpecker, Silviculture, Simulation, Spatially-explicit model, Uncertainty, Wood thrush

FOREST DECISION MAKING UNDER UNCERTAINTY: ADAPTIVE MANAGEMENT FOR THE CONSERVATION OF BIRD POPULATIONS ON A NATIONAL WILDLIFE REFUGE

by

CLINTON THOMAS MOORE

B.S.F.R., The University of Georgia, 1982

M.Stat., North Carolina State University, 1986

A Dissertation Submitted to the Graduate Faculty of The University of Georgia in Partial Fulfillment of the Requirements for the Degree

DOCTOR OF PHILOSOPHY

ATHENS, GEORGIA

2002

FOREST DECISION MAKING UNDER UNCERTAINTY: ADAPTIVE MANAGEMENT FOR THE CONSERVATION OF BIRD POPULATIONS ON A NATIONAL WILDLIFE REFUGE

by

CLINTON THOMAS MOORE

Approved:

Major Professor:	Michael J. Conroy
Committee:	Robert J. Cooper Richard F. Daniels James T. Peterson H. Ronald Pulliam

Electronic Version Approved:

Gordhan L. Patel
Dean of the Graduate School
The University of Georgia
May 2002

DEDICATION

I dedicate this work to Joan and Truman, who instilled in me a love for inquiry and for the natural world; to Scott and Timothy, who I hope will one day inherit, perhaps in small part by my contributions, a planet more mindful of its ecological capacity; and to Mary, who shares her strength, wisdom, and optimism with me each day.

ACKNOWLEDGMENTS

This work would not have been possible without the help of many individuals and benefactors. I thank faculty members Bruce Beck, Robert Cooper, Richard Daniels (University of Georgia Warnell School of Forest Resources), and Ronald Pulliam (University of Georgia Institute of Ecology) for their review of my proposal and their helpful input into my research. Kevin Boston (Carter Holt Harvey Forests, New Zealand) supplied me background information and advice on heuristic and classical optimization procedures, and he provided me a much-needed introduction to geographic information systems. James Peterson's (U.S. Geological Survey, Georgia Cooperative Fish and Wildlife Research Unit, Warnell School of Forest Resources) work with Bayesian belief networks gave me another perspective on making resource decisions under uncertainty.

Bruce Lubow (U.S. Geological Survey, Colorado Cooperative Fish and Wildlife Research Unit) provided comments and suggestions regarding model optimization under program ASDP. Roger Lowe (Warnell School of Forest Resources) provided substantial assistance in the development of digital maps and the spatial database.

I received invaluable assistance from personnel at the Piedmont National Wildlife Refuge. In particular, I would like to thank Dean Metteaeur, John Mason, and Greg Walmsley for technical assistance and information, and Ronnie Shell and Carolyn Johnson for their enlightening discussions on Refuge management.

My student colleagues, Carole Anderson, Laura Dawood, and Larkin Powell, collected much of the data used in my models, and they offered me vital insight on sampling procedures, Refuge management practices, and the general nature of the Refuge. I extend special thanks to colleague Todd Plummer, who worked tirelessly and unselfishly to provide me with data and estimates that made my modeling efforts possible, and who provided comic relief in the darkest hours of project report preparation.

Funding for my research came from the U.S. Environmental Protection Agency STAR Fellowship Program, from the University of Georgia University-Wide Assistantship Program, and from a teaching assistantship provided by the Warnell School of Forest Resources. I also received assistantship and travel support from the Georgia Cooperative Fish and Wildlife Research Unit through a project (Research Work Order 34) funded by the U.S. Geological Survey Cooperative Research Units Center and the U.S. Fish and Wildlife Service, Region 4. The University of Georgia Stoddard-Burleigh-Sutton Award defrayed costs of books and supplies. I am indebted to my employer, the U.S. Geological Survey Patuxent Wildlife Research Center, which supported the final months of my research. William Kendall (Patuxent Wildlife Research Center) sponsored travel to meetings where I presented intermediate results of my research. Sue Anthony and Diane Pritchett (Georgia Cooperative Fish and Wildlife Research Center), two of the most affable, competent professionals with whom I have ever had the pleasure to work, handled the overwhelming details of my funding and plucked me from the administrative flames several times.

I thank my advisor, Michael Conroy (Georgia Cooperative Fish and Wildlife Research Unit, Warnell School of Forest Resources), who provided guidance and served as a valued mentor throughout this difficult project. His advice, both technical and nontechnical, helped me to navigate around a seemingly unending series of obstacles. He worked on my behalf to secure the financial support I needed, and I am grateful for the professional development opportunities he provided me.

Leaving a comfortable professional job, moving my family, and adjusting to life as a “nontraditional” student (I was twice the age of my first classmates in 1997) was personally anguishing for me, and I might have never come through the ordeal without the many friends and family members who offered their support and encouragement. The Bruggink, Kendall, and Querin families kept the lines of communication open and were always there to hear my tired sob stories. The 402 Lab (“Commiseration Central”) hosted a number of mood-altering discussions with Matt, Jennifer, Todd, Nathan, Craig, Kim, Chris, Jay, Carole, Tom, and other colorful Warnell characters. My family, immediate and extended, made sure that I was included in birthday celebrations, beach trips, and reunions, and they were understanding on the many occasions that my work prevented my participation. My parents, Joan and Truman Moore, were always there for my family when my travel schedule or other work requirements left the kids without care. Though my parents have been eager to see me finish, they never demonstrated impatience with the slow pace of my work and have only inquired how they could help.

I watched my children, Scott and Timothy, grow from their pre-K’s into their pre-teens over the course of my study. As they have matured, they too have shown extraordinary patience and understanding in the demands of my class work and research.

I thank them for the time they have given me, and I look forward to being more preoccupied with board games, camping, movies, and baseball.

My last but deepest expression of gratitude goes to my wife, Mary, who made my program of study possible. Mary maintained the house, nurtured the kids, worked every day as a professional statistician, volunteered in the school, and earned a teaching certificate while my work stumbled forward. She has been my most fervent supporter and has always believed in me and my abilities, especially when I could not. Though I cannot fully appreciate the full depth of the sacrifices that Mary made on my behalf, I am nevertheless grateful for her strength and empathy.

TABLE OF CONTENTS

Page

APPENDICES

CHAPTER 1

INTRODUCTION

During the past decade in North America, many government agencies with jurisdiction over natural resources adopted an ecosystem approach to guide their management decisions (Congressional Research Service 1994, Interagency Ecosystem Management Task Force 1995, Brown and Marshall 1996, Keiter 1998, Malone 2000, Nelson et al. 2000). Under this approach, an emphasis on the preservation of ecosystem function and integrity replaces the traditional management focus on maximum return of individual resource commodities, such as wood fiber, food crops, cattle forage, fish, or game (Grumbine 1994, Keiter 1998). Such conventional, rigid control of natural resource systems is thought to reduce system resiliency and induce long-term declines in ecological goods and services (Holling and Meffe 1996). Decision-making tools that were once adequate under the traditional paradigm of resource management are now largely inappropriate, as they often fail to take into account complexities faced by managers of ecosystems (Rauscher 1999). Among these complexities include temporal and spatial dynamics in populations, landscape heterogeneity and scale dependencies, response uncertainty, and multiple dimensions of ecosystem health (Christensen et al. 1996).

Forest management is one such area in which the traditional extraction of renewable resources has been intensively and critically reviewed in an ecosystem context

(Hansen et al. 1995, Carey and Curtis 1996). The controversy surrounding management of forests in the Pacific Northwest may be the most familiar example (Caldwell et al. 1994). There, the population status and habitat needs of a threatened species, the northern spotted owl (*Strix occidentalis caurina*), placed into question the customary forest values and management approaches used in that system (Forsman et al. 1984, Doak 1989, Thomas et al. 1990, Bart and Forsman 1992, Murphy and Noon 1992). In the southeastern U.S., maintaining habitat for an endangered species, the red-cockaded woodpecker (*Picoides borealis*), has profound implications for the conduct of commodity-oriented silviculture, including determination of species composition, rotation length, stocking density, intermediate treatments, and placement and size of cuts (Ligon et al. 1986, Jackson 1994, Wigley et al. 1999).

In circumstances where forest resource extraction is not a primary objective, for example, within the U.S. National Wildlife Refuge System (U.S. Fish and Wildlife Service 1976), resource managers may nevertheless face challenges in choosing management actions that meet ecosystem objectives. For example, recovery of an endangered species may be a primary concern of forest management, but not an exclusive concern. In particular, if actions oriented toward recovery of the species are thought to be at least partly detrimental to other species, then the manager must choose an appropriate level of trade-off among species population objectives. Even then, the optimal choice of actions may not be obvious as responses by populations to forest management actions are typically highly uncertain (Figure 1).

The ability to make optimal resource decisions often eludes managers because practically all natural systems are characterized by uncertainty, complexity, and dynamic behavior. Instead, decisions are frequently made with a degree of risk-aversion and with some unknown opportunity cost (Walters and Holling 1990), and progress is rarely made toward the resolution of uncertainty. Worse, uncertainty may not even be acknowledged, and decisions are then made under unreasonable or unverifiable assumptions about how the managed system operates (Conroy 1993).

A more desirable management approach would be one that provides decisions that are optimal with respect both to the resource conditional on the current degree of uncertainty and to the collection of information that reduces this uncertainty. The objective of my research is to apply this adaptive approach to the development of a decision-making framework for management of a forest. Here, silvicultural actions are carried out on a wildlife refuge where growth of the red-cockaded woodpecker population is but one goal of management. Refuge managers are also charged with the provision of habitat for other wildlife species, habitat which may be degraded or destroyed by actions designed to benefit the woodpecker. The primary feature of this decision framework is the recognition and accommodation of uncertainties regarding responses by wildlife populations to management. Foremost is the uncertainty about the degree of benefit received by woodpeckers and the degree of harm inflicted on other species as a result of woodpecker-oriented silvicultural actions. The decision framework also features stochastic, spatially-explicit population models as woodpecker population dynamics are thought to strongly depend on spatial distribution of both habitat and woodpeckers (Walters 1991, Thomlinson 1995, Letcher et al. 1998, Walters et al. 2002). Outputs of predictive models are measurable in the field. Thus, data collected regularly in the forest provide the means to assess progress toward population objectives and to validate decision-making models.

ELEMENTS OF MAKING AN INFORMED DECISION

Three elements comprise any approach to making optimal decisions for a managed system: a formal statement of one or more objectives, a set of available decision alternatives, and a model that relates the state of the system, each decision action, and the objective function (Walters and Hilborn 1978; Williams 1982, 1989; Figure 2). Specifically, the model predicts response by the objective function to alternative decision actions for any given system state. As many decision problems in natural resource management involve recurring decision opportunities through time, the model must also describe the dynamics of the system; in particular, the model should forecast future system states in response to decision actions and current states (Williams 1982, 1989). Following specification of all three elements, and given a measured state of the system (e.g., via a monitoring program), an optimal, or nearly optimal, decision can be identified for either the case of a static or dynamic system (Williams 1989).

If decision making is to explicitly address uncertainty regarding the system model, then an additional element, a model set, is required. Uncertainty may be manifest in terms of structural uncertainty, where two or more models provide alternative, structurally distinct representations of the system (e.g., Williams 2001; Figure 3), or in terms of parametric uncertainty, in which a continuum of an infinite number of models portrays uncertainty about the specific value of a model parameter (e.g., McAllister and Kirkwood 1998). In either case, a discrete or continuous distribution of subjective probability, respectively, is assigned to all models in the model set.

For dynamic systems, the probability distribution may be updated through time by comparing alternative model predictions of the system state to the measured conditions (Figure 4). In this case, the monitoring program serves two purposes: for choosing the appropriate state-specific optimal action at each decision opportunity and for periodically updating belief in each alternative model or parameter value (Williams and Nichols 2001). Thus, information is accumulated through time as decisions are made and the system response is monitored. Because optimal decisions are conditional on this "information state" as well as on the physical system state, the (physical) state-specific decision policy adapts over time as information is accumulated. That is, even if information is completely lacking (i.e., noninformative prior probabilities are placed on each model or each possible parameter value), an optimal state-specific decision policy may nevertheless be attained, with the recognition that the collection of information will only contribute toward management objectives and thus improve decision making (Conroy and Moore 2002). In this way, the collection of data helps to inform future decision making.

MANAGEMENT ON THE PIEDMONT NATIONAL WILDLIFE REFUGE

I conducted my work on the U.S. Fish and Wildlife Service Piedmont National Wildlife Refuge (hereafter, "Refuge") in central Georgia, USA (Figure 5). The Refuge hosts the largest population of red-cockaded woodpeckers that occur in the Piedmont physiographic province of the southeastern U.S. This population, approximately 40 breeding groups in 2000 (U.S. Fish and Wildlife Service Piedmont National Wildlife Refuge, unpublished data), and a smaller population of 18 groups on the adjoining USDA Forest Service Oconee National Forest (unpublished report, "Management Indicator

Species Population and Habitat Trends Chattahoochee-Oconee National Forests," U.S. Forest Service Southern Region, 2000) were designated a secondary core population for recovery purposes (U.S. Fish and Wildlife Service 2000). Under the Endangered Species Act and related directives, refuges are mandated to maintain and improve habitat for red-cockaded woodpeckers where they do or could occur (U.S. Fish and Wildlife Service 1998).

Consequently, management of the loblolly pine (*Pinus taeda*) forest on the Refuge is oriented toward the creation and retention of suitable habitats for nesting and foraging. These habitats are characterized by open stands of mature pines, herbaceous understories, and nearly absent midstories (Ligon et al. 1986, Walters 1991, James et al. 2001). These conditions are especially critical in nesting habitats (Hovis and Labisky 1985, Loeb et al. 1992), in which clusters of very old (≥80 years) living trees are preferred for nest and roost cavities (Hovis and Labisky 1985, DeLotelle and Epting 1988, Walters 1991). To this end, aggressive programs of prescribed burning and midstory reduction are carried out on the Refuge (unpublished report, Habitat Management Plan, U.S. Fish and Wildlife Service, Piedmont National Wildlife Refuge, 1982; hereafter "Refuge Habitat Management Plan").

At the same time, closed canopy and shrubby understory conditions that occur in both pine and hardwood stands throughout the Refuge provide nesting habitat for a number of terrestrial and arboreal species, including the wood thrush (*Hylocichla mustelina*). Wood thrushes, like several other species of forest-interior neotropical migratory birds, are of concern to management agencies because of their range-wide population decline (U.S. Fish and Wildlife Service 1995). The Refuge is also charged with managing for this group of birds (U.S. Fish and Wildlife Service 1988).

Thus, improvement of habitat conditions for one trust species apparently occurs at the cost of habitat for the other. The term “apparently” is meaningful here because outcomes of management for these species are not certain. A previous investigation of wood thrush response to woodpecker-oriented management found no effect of management on many vital rates of wood thrushes; in fact, and counter intuitively, positive responses were detected in many demographic parameters (Powell et al. 2000). Though these results do not agree with naturalist intuition about the wood thrush, they are not entirely unexpected given the high level of measurement error experienced in the study and the moderate nature of the treatments (Powell et al. 2000). Therefore, managers should not interpret the failure to detect anthropogenic effects as a demonstration of absence of such effects. Consequently, despite the effort and expense in conducting crucial experiments, a course of silvicultural action may not be obvious to managers. This scenario is often the case in conservation management, where decisions cannot be delayed until all uncertainties have been resolved.

Uncertainty over wood thrush response aside, management solely for woodpecker objectives is not clear-cut, given the complexity in population dynamics thought to exist for this species. The woodpecker’s elaborate social structure, its limited dispersal, and its requirement for specific, ephemeral, unfragmented (Conner and Rudolph 1991) habitat suggest that woodpecker populations are sensitive to spatial arrangements of treatments, stochastic disturbances, and woodpeckers themselves (Walters 1991). Thus, decision models for woodpecker management likely require some form of spatially-explicit representation (Pulliam et al. 1992, Dunning et al. 1995, Letcher et al. 1998). Consequently, optimal decision making becomes extremely difficult as the set of management options explodes into an overwhelming number of alternative spatial arrangements of forest actions.

Thus, managers of the Refuge's wildlife resources face the difficult challenge of choosing the best course of management when satisfaction of multiple objectives appear to call for completely different strategies, when responses by the populations to management are not clearly known, and when complexity in population dynamics admits an inconceivably large set of management options. Unfortunately, such challenges are not unique in conservation management, and managers require decision tools that will help them confront these difficulties.

RESEARCH OBJECTIVES

My research had four objectives. First, I wanted to develop one or more prediction models for the long-term response of habitat and bird populations (woodpecker and wood thrush) to spatially-distributed forest management. These models would be developed from data in the published literature, from data collected on-site at the Refuge, or from reasoned, intuitive guesses. By formulating a set of alternative, plausible competing models, my intent was to place reasonable bounds on uncertainty about forest and bird responses.

My second objective was to identify optimal courses of management for bird population objectives, conditional on model choice. Here I wanted to find how to distribute forest actions over space and time to separately maximize abundance of woodpecker groups, abundance of wood thrushes, and a composite abundance measure of both species. Results of these analyses would indicate sensitivity of the optimal strategy to choice of the decision model and objective function.

Thirdly, I wanted to find optimal decision strategies for each response with explicit recognition of uncertainty among all models. I wished to demonstrate that even when critical data are lacking for model building, decision making can nevertheless proceed provided that uncertainty is explicitly acknowledged. This analysis would also indicate the expected gain in value of each management objective brought about by the reduction of uncertainty.

Finally, based on my modeling and analysis results, I wanted to provide recommendations to the Refuge for conducting habitat and bird monitoring. Such a monitoring program is key to making management truly informative and adaptive.

CHAPTER 2

LITERATURE REVIEW

SOURCES OF UNCERTAINTY

Four sources of uncertainty often frustrate decision making for natural resource objectives (Nichols et al. 1995, Williams 1997, Charles 1998). As stated earlier, a decision model describes how a system and the objective function respond to a given management action from a starting system state. The first source of uncertainty is that of partial observability, in which the system state is not completely observed or is measured with error, for example, the estimation of forest understory density based on a sample of sites. Partial observability implies that decisions are made based on apparent rather than real states of the system, and that assessment of decision impacts on the system or on the objective is imprecise.

The second source is environmental uncertainty, which encompasses all forms of variability intrinsic to the biotic and abiotic components of the system, principally demographic and environmental stochasticity. Partial controllability, the third source, reflects the degree to which a realized decision action departs from the intended action, for example, how closely a prescribed fire burns its intended coverage at its intended intensity.

The last source, structural uncertainty, is uncertainty regarding the fundamental nature of the system response to the management action. As previously discussed, this source of uncertainty can be represented as uncertainty about a specific parameter value or in the model's structural form itself. Together, the four sources of uncertainty imply that the system response to a given decision action inherits a probability distribution, obscured by partial observability, that is unique to each uncertain system model or parameter value (Williams 2001, Williams and Nichols 2001).

STRUCTURAL UNCERTAINTY, MODES OF DECISION MAKING, AND ADAPTIVE MANAGEMENT

All sources of uncertainty described above are troublesome for decision makers, but perhaps none more so than structural uncertainty. Faced with uncertainty among multiple models or within a continuum of possible values for a parameter, managers rely on one of several modes of decision making (Peterman and Anderson 1999). The first is to manage as if no structural uncertainty exists, either through ignorance or more consciously through explicit denial of uncertainty. For example, one very common way of developing a decision model is to fit a set of data to each of several competing models, then select a winning model based on some goodness-of-fit criterion (Collie and Walters 1993). Behaving as though this model represents truth can have serious consequences for the decision maker, the agency, and the resource (Conroy 1993, Radomski and Goeman 1996). The manager may rationalize this mode of decision making by arguing that whatever uncertainty exists is likely to be small or that unexpected responses by the system to an action can be corrected or compensated for in the next decision cycle. Of course, without consideration of competing models, these speculations cannot be verified and the manager may never know how far from optimal a course of management may be under a single model (Pascual et al. 1997). Worse, if after a series of decisions it finally becomes clear that the model is unsuitable, the manager is left with no knowledge about the relative performance of models not considered, only that the chosen model was unsatisfactory.

Another mode of decision making acknowledges uncertainty, but it emphasizes a policy that averts risk toward one principal resource objective. In conservation settings, risk is a function of the cost of an undesirable management outcome and its probability of occurrence (Mowrer 2000). A manager facing a decision problem for which several models are all plausible may arbitrarily emphasize the model that forecasts the greatest likelihood of unacceptable resource loss under certain decision actions. Conditional on this model selection, the manager then chooses the action that is expected to incur the least harm to the resource. Such precautionary approaches to decision making are commonly endorsed for endangered species management and biodiversity conservation (Myers 1993). Here, the costs of action may be high, but appropriate selection of a resource utility (benefit) measure or selection of a sufficiently pessimistic management model will nevertheless tend to call for these actions to minimize chance of extirpation or extinction. Of course, this approach invites controversy because the issue of whether recovery goals can be met at less cost is never addressed (Lindley 1985:187-189, Bodansky 1991, Schweder 2001).

In the worst cases, uncertainties and stakes may be so great that a decision-making paralysis sets in. Often in such circumstances, the decision that is ultimately made is one that either maintains a status quo or is politically expedient (Walters 1986:30-35, Walters and Holling 1990).

Alternatively, the decision maker can use principles of adaptive management and adaptive optimization to explicitly confront uncertainty among system models (Walters and Hilborn 1978, Walters 1986, Williams 1996). These approaches permit optimal decision making under simultaneous consideration of all uncertain models, and no model must be subjectively favored over any other as is the case under the other modes of decision making. The primary difference between this and non-adaptive approaches is that the system is measured with respect to both its physical state and an information state. The information state simply reflects the current degree of belief or credibility, expressed as a probability, in each of the alternative models or parameter values. To the extent that the distribution of credibility weights can be adjusted from an even distribution among models to one that places all weight on one of the models, management decisions can be continuously improved, as measured by progress toward resource objectives. The principal means by which information states are adjusted is through the collection of monitoring data and by comparing the observed state of the system to predictions generated by each of the models. Bayes' Theorem is the formal mechanism that carries out these probability adjustments given the model predictions (Lindley 1985). The continuous cycle of decision making, prediction, data collection, and information updating constitutes adaptive management (Johnson et al. 1993).

Optimization under model uncertainty can be approached in either of two ways. In a passive approach, the information state (belief in each model) is assumed to stay constant through time (Walters and Hilborn 1978, Williams 1997). Therefore, the optimization framework does not anticipate change in the information state in response to management actions. Of course, the information state can be updated by application of

Bayes' rule following any management action, so in implementation, the information state indeed changes through time. The optimization can be repeated at the next decision cycle, at which time the updated information state is again assumed to remain fixed. Thus, while information is nevertheless accumulated through management by this approach, it is not actively pursued; its collection occurs only as an unplanned by-product of management.

In contrast, an active approach to optimization anticipates change in the information state as a consequence of management decisions (Walters and Hilborn 1978, Williams 1997). Because the information state can be changed through management, the selection of optimal decisions for the resource objective depends on the current level of uncertainty and how effectively decisions can be used to reduce uncertainty. Thus, information is actively pursued, and the reduction of uncertainty becomes an implicit co-objective of management. When uncertainty is high, optimal decision actions are those that gently prod the system to elicit information. As uncertainty decreases and evidence accumulates toward a single model or parameter value, optimal decisions reflect the increasing influence of that model or value.

POPULATION BIOLOGY OF THE RED-COCKADED WOODPECKER

The red-cockaded woodpecker is a non-migratory territorial cavity-nester (Jackson 1994). Once widespread through the southeastern U.S. (Jackson 1994), the woodpecker was declared endangered in 1970 (Ligon et al. 1986). The species is endemic to the longleaf pine (*P. palustris*)-wiregrass (*Aristida* spp.) ecosystem of the coastal plain region, but isolated populations also occur in loblolly or shortleaf (*P. echinata*) forests as far north as the Cumberland Plateau and southeastern Virginia

(Jackson 1994). The longleaf-wiregrass community, historically maintained by frequent wildfire, provides the conditions preferred by woodpeckers for foraging and nesting: a sparse overstory of mature pine, a relatively absent midstory, and an herbaceous understory (Ligon et al. 1986, Jackson 1994). These habitats began to disappear in the 19th century as fire was suppressed and as woodlands were converted to agriculture and short-rotation wood fiber production (Ligon et al. 1986, Jackson 1994). Successful management of the woodpecker largely depends on the effectiveness of restoring these habitats through silvicultural means, primarily regeneration cutting, thinning, midstory removal, and prescribed burning (Ligon et al. 1986, Jones 1993, U.S. Fish and Wildlife Service 2000). Publicly-owned lands, on which management objectives do not focus on timber revenues, are the main target of recovery efforts (U.S. Fish and Wildlife Service 2000).

Red-cockaded woodpeckers excavate nesting and roosting cavities in the heartwood of mature living pines (DeLotelle and Epting 1988). Very old trees (usually ≥ 80 years) are required for cavity excavation, as they generally provide a suitably soft and suitably large volume of heartwood at a suitable bole height (6-15 m; Walters 1991, U.S. Fish and Wildlife Service 2000). Furthermore, cavity excavation is an arduous (1-15 yr) and hazardous undertaking for a woodpecker (Walters 1991, U.S. Fish and Wildlife Service 2000). Thus, adequate cavity habitat is quite ephemeral and difficult to generate by natural means. For this reason, artificial means of cavity supplementation developed by Copeyon (1990) and other workers were heralded by woodpecker managers as a breakthrough for woodpecker recovery efforts (Costa 1995).

Woodpeckers form and maintain family groups within clusters of cavity trees (Walters 1991, Jackson 1994). Clusters either are actively populated by woodpeckers or are inactive, available for occupation by dispersing woodpeckers. Fledglings of both sexes typically disperse in their natal year to seek breeding opportunities in nearby clusters. However, some fledglings, most often males, remain with the family group and help with the upbringing of subsequent broods. In this way, male helpers are poised to replace the breeding male should he die or abandon the group. For both males and females, dispersals are fairly local (median distance <4 km; Walters 1991).

Though the habitat needs of the woodpecker are well known in a general sense, there is disagreement and uncertainty regarding specific guidelines for habitat management. In 1985, the U.S. Fish and Wildlife Service published quantitative criteria for the establishment and maintenance of foraging habitat around each cluster (U.S. Fish and Wildlife Service 1985). These guidelines prescribed a minimum of 51 ha of preferred foraging habitat made available within 800 m of the cluster. According to the guidelines, preferred foraging habitat comprised trees at least 30 years old with 40% or more of the basal area in trees at least 60 years old. A number of researchers have taken issue with these guidelines and others modeled after them (Jackson 1994). Ligon et al. (1986) pointed out that the guideline values were derived from an analysis of a single, small population of birds occurring in superior habitat. They believed that these values were insufficient in light of the widespread distribution of woodpeckers across a variety of habitats, and they therefore recommended more restrictive area and age composition criteria. Beyer et al. (1996) found no association between amount of foraging habitat and woodpecker productivity on a national forest in Florida. Similarly, Wigley et al. (1999)

detected no habitat-productivity association on intensively-managed private forests in Louisiana. In the North Carolina Sandhills, Davenport et al. (2000) found a counterintuitive negative relationship between productivity and tree diameter, a finding that they attributed to an unmeasured interaction, to an unexplored threshold effect, or to incomplete sampling of the foraging area. James et al. (2001) concluded that not only were the attribute guideline values inappropriate, the choice of attributes themselves was inappropriate. They recommended expression of guidelines in terms of forest and understory structure (i.e., tree age distributions and vegetation composition) rather than in terms of total quantities of stems or basal area (James et al. 2001).

Increasing foraging habitat may be effective in increasing fledgling productivity or adult survival, and therefore, total population size (Heppell et al. 1994). However, availability of nest cavities apparently limits the number of breeding pairs in many populations (Walters 1991, James et al. 1997), so manipulation of foraging habitat alone may be ineffective in increasing number of family groups (Walters 1991, Heppell et al. 1994). The pool of helpers simply expands or contracts with changes in survival and productivity conditions, but because of cavity limitation, the total population of family groups remains mostly stable (Walters 1991, Heppell et al. 1994). Thus, the principal means by which a woodpecker population is grown is through the artificial or natural creation of new nesting clusters, not through means that increase survival or productivity. Furthermore, successful occupation of clusters is dependent on degree of isolation from other clusters and habitat fragmentation between clusters (Thomlinson 1995).

Thus, predicting the effects of management on red-cockaded woodpecker populations is not straightforward. Woodpeckers appear sensitive to forest fragmentation

(Conner and Rudolph 1991), and the very specific and ephemeral nature of the required nesting habitat, the family group structure of the population, and the limited dispersal of fledglings (Walters 1991) suggest that successful habitat planning for woodpeckers must be carefully coordinated through time and over space, perhaps at the resolution of individual forest stands.

For at least two reasons, prescribed burning is an important management tool for red-cockaded woodpeckers. First, burning retards the encroachment of midstory hardwood vegetation within woodpecker clusters (Jones 1993). Cluster use has long been associated with reduced hardwood midstory encroachment (Van Balen and Doerr 1978, Hovis and Labisky 1985, Loeb et al. 1992). Midstory encroachment may facilitate access to the nest cavity by avian, mammalian, and reptilian nest predators (Loeb et al. 1992). Encroachment may also encourage cavity expropriation by avian and mammalian kleptoparasites, such as pileated woodpeckers (*Dryocopus pileatus*) and flying squirrels (*Glaucomys volans*) (Conner and Rudolph 1989).

Secondly, prescribed burning promotes existence of an herbaceous understory (Masters et al. 1996, James et al. 1997). An important dietary source for woodpeckers are arthropods found in understory vegetation (Hess and James 1998). Productivity and fledgling survival may depend on availability of these diverse food sources (James et al. 1997).

The U.S. Fish and Wildlife Service identified a recovery goal of 96 woodpecker groups on 7776 ha of pine habitat at the Refuge (U.S. Fish and Wildlife Service 1998, 2000). According to the Service's projection of 10% annual population growth under this recovery plan (U.S. Fish and Wildlife Service 2000), the goal should be reached in

10 years. The pine habitat quantity in this goal reflects the intent of Refuge managers to permit hardwood succession to eventually comprise 40% of all forested area (Refuge Habitat Management Plan), or approximately 5654 ha of the 14,136-ha Refuge. To meet this population goal, the recovery guidelines call for growing-season prescribed burning on a 3-5 year basis (U.S. Fish and Wildlife Service 2000).

However, the recovery guidelines also call for an ecosystem approach to management, in which management mimics natural processes to the extent possible, does not focus on single-species objectives, and operates in a landscape rather than a patch-oriented context (U.S. Fish and Wildlife Service 2000). Despite the obviously important role of public lands in woodpecker recovery objectives, this mandate may pose a troublesome dilemma for woodpecker management at the Refuge. Longleaf-wiregrass systems in the southeastern coastal plain were historically maintained by lightning-induced fires occurring on a 1-3 year basis (Frost 1998). Therefore, single-species management oriented toward the red-cockaded woodpecker could be justified in this case as the woodpecker is arguably an indicator/keystone/umbrella species in that system (U.S. Fish and Wildlife Service 2000). However, Piedmont loblolly forest communities are more structurally diverse than longleaf-wiregrass systems and experienced naturally-induced fires less frequently (4-6 years; Frost 1998). In the absence of fire, disturbed areas in the Piedmont quickly succeed to hardwood climax conditions (Johnston and Odum 1956). A hardwood forest was predominant in the eastern half of the Georgia Piedmont prior to European settlement (Nelson 1957). Although the woodpecker was once widespread throughout the southeast in a variety of pine ecosystems (Jackson 1994), it is not unreasonable to expect that woodpecker populations followed cycles of extirpation and resettlement in areas that are characterized by occasional patch disturbance, relatively infrequent fire, and rapid rates of hardwood succession. In other words, the red-cockaded woodpecker may be an ecological indicator species for only a transitory state of the Piedmont forest. Therefore, a single-species focus may be less justifiable in the Piedmont than in the coastal plain, at least from an ecosystem perspective. Furthermore, the application of high management inputs to maintain the forest in an artificial state of hardwood suppression seems to conflict with the principles of ecosystem management. Therefore, relative to management in coastal plain environments, conservation objectives for the Refuge may not be as clearly or unanimously established.

POPULATION BIOLOGY OF THE WOOD THRUSH

The wood thrush is a neotropical migratory bird of eastern North American forests (Roth et al. 1996). Beginning in early April, wood thrushes arrive in Georgia (Weaver 1949) from their wintering grounds in the Yucatan Peninsula southward to Panama (Roth et al. 1996); breeding commences at the Refuge by early May (Powell 1998). Wood thrushes are believed to be monogamous within the breeding season but seldom are found with the same mate in successive breeding seasons (Roth et al. 1996). Pairs typically attempt more than one brood each year and often raise two broods (Weaver 1949, Brackbill 1958, Roth et al. 1996). Wood thrushes forage predominantly for invertebrates in leaf litter, but the diet includes fruits, especially in the late summer and fall (Roth et al. 1996). Wood thrush nests are constructed of a variety of natural and man-made materials interwoven in mud or rotting vegetation (Weaver 1949, Roth et al.

1996, Powell 1998). Nests are located in shrubs or in the lower branches of trees, usually <6 m from the ground (Roth et al. 1996).

Archetypical breeding habitat of the wood thrush is the interior of mature, closed-canopy, deciduous or mixed composition woodlands (Weaver 1949, Hamel et al. 1982, Roth et al. 1996). Some authors, observing that wood thrush territories often incorporate forest edges, classify the bird as an interior-edge species (Whitcomb et al. 1981). The wood thrush requires a sparse to moderate deciduous understory for perching and nest placement; an absent or sparse ground cover facilitates foraging (Roth et al. 1996). Nest sites are often found in bottomlands, upland draws, and other mesic habitats that provide the necessary substrate for foraging and nest-building activities (Weaver 1949, Bertin 1977, James et al. 1984, Roth et al. 1996, Powell 1998).

Populations of wood thrush are believed to have declined throughout their range since the early 1980s (Peterjohn et al. 1995). As is the case with many other declining populations of forest-dwelling neotropical migrants, researchers and managers are uncertain whether declines stem from loss or degradation of breeding habitat, wintering habitat, migration habitat, or all three (Sherry and Holmes 1992, 1995). Many forest interior migrants are thought to be sensitive to forest fragmentation and patch isolation of breeding areas (Whitcomb et al. 1981, Temple and Cary 1988, Hansen and Urban 1992). Lynch and Whigham (1984) observed a negative association between densities of wood thrush and a measure of patch isolation (distance to nearest forest ≥50 ha). Similarly, Robbins et al. (1989) found that forest patch size and patch isolation interacted with respect to detection of wood thrushes: detection rate was consistent in different patch sizes when patches were minimally isolated (≥67% forest ≤2 km of patch) but was negatively correlated with patch size when patches were greatly isolated (≤33% forest ≤2 km of patch). Keller et al. (1993) were more likely to detect wood thrushes in wider rather than narrower riparian forest buffers. Small patches may occur as population sinks that are continually replenished by dispersing or displaced wood thrushes from nearby source habitats (Pulliam 1988). Brawn and Robinson (1996) attributed low rates of wood thrush productivity in a highly fragmented Wisconsin landscape to high rates of nest predation and brown-headed cowbird (*Molothrus ater*) nest parasitism. The range-wide population decline and degradation of breeding habitat prompted Partners in Flight to identify the wood thrush as a species of high management concern in the southeastern U.S. (Hunter et al. 1992).

Though much evidence suggests that wood thrush breeding populations are negatively affected by increased fragmentation of the landscape into forest and non-forest uses (Donovan et al. 1995; but see Friesen et al. 1999), it is less clear whether disturbances and fragmentation brought about by silvicultural practices similarly affect densities and other demographic parameters (Duguay et al. 2001). Silvicultural practices alter canopy cover, understory composition, and other components of wood thrush breeding habitats, and it is reasonable to expect that these practices potentially affect wood thrushes at both the forest stand and landscape scales. Certainly, even-aged regeneration treatments (clearcuts, seed tree cuts, shelterwood cuts; Thompson et al. 1995), which remove all or major portions of the canopy cover, temporarily (12-30 yr) displace local wood thrush populations as late-succession breeding habitats (Johnston and Odum 1956, Meyers and Johnson 1978) yield to unsuitable early-succession habitats (Conner and Adkisson 1975, Dickson et al. 1993, Annand and Thompson 1997). The various forms of uneven-aged management (single-tree selection, group selection, fuelwood harvest; Chadwick et al. 1986, Thompson et al. 1995) affect the forest canopy less severely than even-aged approaches. Thus, wood thrushes may persist in the stand, albeit at reduced densities (Crawford et al. 1981, Chadwick et al. 1986, Annand and Thompson 1997). At the scale of a forest landscape, however, interspersion of clearcuts and other regeneration cuts within the forest appear to not affect densities of wood thrushes in neighboring mature stands (Thompson et al. 1992). Duguay et al. (2001) found no evidence that forests fragmented by various even-aged silvicultural treatments created population sinks for wood thrushes.

A number of intermediate silvicultural practices may take place over the life of a stand (Thompson et al. 1995); those most pertinent to this study are overstory thinning, hardwood midstory reduction, and prescribed burning (Refuge Habitat Management Plan). Few experimental studies have addressed the response of wood thrushes to such periodic treatments. Dickson (1981) speculated that densities of shrub-nesting forest birds are reduced in proportion to amount of understory vegetation removed by burning. Relative to untreated stands, Rodewald and Smith (1998) found sharply reduced densities of wood thrushes in oak (*Quercus* spp.)-hickory (*Carya* spp.) stands where hardwood understory vegetation had been mechanically removed.

Several observational studies investigated intermediate treatments oriented specifically toward red-cockaded woodpeckers and responses by non-target bird species to those treatments (hardwood reduction by dormant-season prescribed burning and by mechanical or herbicidal means). In all studies, and to varying degrees, wood thrushes were less abundant or less common in treated than in untreated stands (Lucas 1994, Brennan et al. 1995, Burger et al. 1998, Raftovich 1998, Hines 1999). In an experimental study of thinning and dormant-season burning in mixed stands managed for red-cockaded woodpeckers, Wilson et al. (1995) found that treated stands, compared to control stands, generally contained fewer individuals of hardwood-dependent forest-interior bird species; however, the wood thrush was not one of their studied species. Block et al. (1995) concluded that woodpecker-oriented management was not necessarily incompatible with management for forest-interior neotropical migrants, and they urged further experimental research into non-target bird response to variations in woodpecker management practices and habitats.

The only experimental investigation of wood thrush demographic parameter response to intermediate forest treatments occurred at the Refuge (Powell 1998, Powell et al. 2000). In this study, control and treated management compartments were compared before and after application of thinning and burning treatments. No effects of treatment were detected on wood thrush densities, adult or juvenile rates of survival (Powell et al. 2000), or daily nest success (Powell 1998). In fact, most estimated treatment-control differences were positive (Powell 1998). Treatments did not affect juvenile dispersal distances, but adults in treated areas were less likely to disperse within the breeding season than those in control areas (Powell 1998). Wood thrushes on treated areas more strongly selected hardwood habitats and avoided pine habitats following the treatment than before, whereas habitat preferences on untreated units were similar before and after treatments (Powell 1998). Powell (1998) speculated that the relatively modest impacts of the treatments, the mobile nature of wood thrushes, and the ability of wood thrushes to employ a somewhat wide range of nesting and foraging habitats may explain the failure to detect treatment effects in this study population. Furthermore, many estimates of treatment effect were highly uncertain despite intensive field efforts to control and reduce sources of variability (Powell et al. 2000). Powell et al. (2000) concluded that management oriented toward red-cockaded woodpecker habitat creation and maintenance, as currently practiced on the Refuge, does not negatively affect wood thrush population density or growth rate. However, recognizing the high variability in many of the measured parameters, they nevertheless recommended an adaptive course of management that incorporates a program of habitat and population monitoring (Powell et al. 2000).

MODELING BIRD POPULATION RESPONSES TO HABITAT MANAGEMENT ALTERNATIVES

Models linking habitat management decisions to bird population outcomes fall into three general classes. The first class employs an observed statistical (or speculated) link between habitat conditions and bird occurrences or abundances to make predictions about bird response to unobserved future conditions (Verner et al. 1986, Buckland and Elston 1993, Hepinstall and Sader 1997). The advantage of these models is their easy linkage to geographic information system (GIS) habitat mapping resources and their resultant spatial detail. Their disadvantage is their underlying premise that population abundance is a monotonic function of habitat quality, independent of local population dynamics (Van Horne 1983, Pulliam 1988, Conroy 1993, Conroy and Noon 1996). These models are entirely phenomenological, that is, they lack a mechanistic link that describes how changes in habitat affect population demographic parameters such as survival, productivity, and dispersal. Thus, such prediction models are of dubious reliability under conditions different from the ones under which they were developed (Conroy and Moore 2002).

A second class of models is non-spatially explicit demographic models (Shaffer 1981, Noon and Sauer 1992, Caswell 2001). These models use demographic information on survival and productivity, often specific to different life stages, to project populations through time. The models can be examined for sensitivity to habitat actions that are believed to affect demographic parameters. These models may provide the mechanistic link between habitat and demographic response, but they have limited realism for systems in a spatially heterogeneous landscape. Such models may be entirely suitable for management problems in which the target species is highly mobile and not strongly tied to a specific, patchily-distributed habitat feature.

The third class of models, spatially-explicit population models, integrate spatial representations of the habitat with animal population dynamics (Pulliam et al. 1992, Dunning et al. 1995). Habitat features may vary through time and throughout the landscape according to a management and succession model, and they are linked to elements (individuals, groups, or subpopulations) of an animal population model (Holt et al. 1995). Animals in the model respond both to distribution of habitats and to other animals. As with habitat association models, spatially-explicit population models are easily bridged to GIS resources. Unlike those models however, spatially-explicit population models incorporate survival, recruitment, and dispersal mechanisms. Spatially-explicit population models may be most useful for species that are rare or dependent on patchily-distributed habitat elements. Although such models offer a great degree of ecological realism, they are very difficult to parameterize and uncertainties in the parameters compound to render output of questionable reliability (Conroy et al. 1995).

MANAGEMENT MODELS FOR THE RED-COCKADED WOODPECKER

Models of all three classes have been proposed for managing red-cockaded woodpecker populations. Many investigators postulated or statistically inferred relationships between habitat measurements and a response variable (e.g., woodpecker occurrence, nest site occurrence, nest site activity) to generate predictive habitat association models at the forest stand (Connor and Rudolph 1989, Loeb et al. 1992, Thomlinson 1996, Doster and James 1998, Wigley et al. 1999, Davenport et al. 2000), landscape (Seagle et al. 1987, Azevedo et al. 2000), and regional (Flather and King 1992) scales. The model of Loeb et al. (1992), which associated cluster activity with degree of hardwood midstory encroachment, was developed from data collected at the Refuge.

Reed et al. (1993) and Stevens (1995) used non-spatially explicit population models to estimate effective sizes (N_e) of woodpecker populations. Stevens (1995) acquired demographic data from a population study conducted on the Refuge (Lennartz and Heckel 1987) to parameterize both a deterministic, analytic model and a stochastic simulation model. Based on sensitivity analysis of the simulation model, he concluded that management focused on increasing female adult and fledgling survival was more effective in increasing the Refuge population than actions designed to increase productivity. Constructing artificial cavities, he suggested, would stem off-site dispersal by females (Stevens 1995). Using the data of Lennartz and Heckel (1987) plus additional resighting data collected by the authors, Maguire et al. (1995) also developed a stochastic population viability model. Results of their analysis were equivocal, however. The authors, uncertain over the appropriate set of data to use for survival calculations, found evidence both for population growth and population decline. In the face of this uncertainty, the authors recommended, as did Stevens (1995), that woodpecker management should aggressively pursue measures to reduce fledgling mortality, specifically, through the construction of artificial cavities. In each of the foregoing population modeling exercises, however, all authors pointed out the deficiencies of their models with respect to the complex woodpecker social structure and dispersal dynamics.

In their population model, Heppell et al. (1994) attempted to explicitly take into account male woodpecker social structure. They constructed a deterministic stage-based population model using demographic data collected in the North Carolina Sandhills region. Stages in the model were mutually exclusive social classes of breeding and nonbreeding males. They informally linked their model to management by experimentally adjusting productivity and transition parameters according to how they believed different actions would affect the parameters. This form of sensitivity analysis suggested that hardwood removal in occupied areas and constructing artificial cavities in unoccupied areas were most likely to increase numbers of breeding groups. Other actions (e.g., exclusion of cavity competitors, improving foraging habitat) were believed to enhance average group size rather than increase group numbers.

Letcher et al. (1998) developed an individually-based, spatially-explicit demographic model to investigate woodpecker population persistence under varying degrees of territory aggregation. The authors simulated production, dispersal, pair-bonding, and mortality events of individuals on an artificial, homogeneous landscape. The events were chosen randomly from fixed probability distributions, thus the model featured demographic stochasticity. The principal finding was that small populations were just as persistent as larger ones provided that territories were sufficiently clumped rather than dispersed. The pool of male helpers in highly aggregated territories maintained population stability in the presence of demographic variability (Letcher et al. 1998). Walters et al. (2002) improved on the Letcher et al. (1998) model by incorporating environmental stochasticity in demographic parameters; nevertheless, their findings regarding effects of spatial aggregation of territories were much the same as those of Letcher et al. (1998). However, neither model was linked to a habitat component and therefore did not accommodate habitat variability through time or across space.

In contrast, Hughell (1996) analyzed a spatially-explicit woodpecker population model that did incorporate and respond to habitat variability. Compared to the Letcher et al. (1998) and Walters et al. (2002) models, Hughell's (1996) model was demographically simpler because the population unit of interest was the breeding group, not the individual. His model tracked the status of breeding groups in three artificial landscapes and in one real landscape, each landscape partitioned into 4-ha hexagonal cells. Breeding groups persisted, died, or "budded" into neighboring habitat cells according to stochastic rules moderated by local habitat conditions.

Hughell's (1996) work provided the only example in which dynamic, map-referenced vegetation growth models linked woodpecker population status to alternative management scenarios. As timber production was a co-objective of management in his study, he employed yield equations to project timber volume through time. He estimated stand-scale foraging habitat quality on the basis of locally-measured site index and stand age, and probabilities of breeding group persistence and expansion were governed by quantity of suitable foraging habitat in the neighborhood of the group. Hughell (1996) used his model to optimally schedule stand-level harvests through time, given a degree of trade-off between harvesting volume and foraging resource objectives. Seagle et al. (1987) also presented a stand-based forest decision simulation model, but stand age was evidently the sole habitat variable they modeled. Their model did not explicitly tie woodpecker response to management actions; rather, they evaluated alternative decision scenarios only in terms of suitable woodpecker habitat (defined as area of forest exceeding minimum threshold ages) created.

MANAGEMENT MODELS FOR THE WOOD THRUSH

Demographic models for migratory birds are challenging to develop as the constituent model parameters are difficult to measure in the field (Noon and Sauer 1992). Nevertheless, Noon and Sauer (1992) demonstrated the use of a simple stage-structured population model for wood thrush to gain insight into effective management approaches. Donovan et al. (1995) employed a two-stage wood thrush projection model to distinguish source from sink forested habitats in the midwestern U.S. Substituting locally-measured estimates of productivity and adult survival in the same model, Simons et al. (2000) estimated a wood thrush production surplus (population growth rate > 1) in the Great Smoky Mountains National Park. In similar use of the model of Donovan et al. (1995), Duguay et al. (2001) also concluded that all the forested areas they studied in West Virginia produced a surplus of young available for dispersal.

Habitat models for wood thrush, on the other hand, are relatively simple to develop, are easily linkable to maps of vegetation and physiography, and thus have received greater attention by researchers. Point counts of wood thrushes have been correlated to forest stand structural variables (DeGraaf et al. 1998), edge types (Mancke and Gavin 2000), and to vegetation characteristics and landscape metrics (Fauth et al. 2000, Penhollow and Stauffer 2000). Similarly, binary detection/non-detection data have been correlated to physiographic features (Dettmers and Bart 1999), to habitat and topographical characteristics (Simons et al. 2000), and to landscape metrics and vegetation attributes (Mitchell et al. 2001). Hoover and Brittingham (1998) used measured vegetation features to distinguish nest sites from randomly-chosen non-nest sites. Powell et al. (2000) estimated wood thrush density in each of several forest management compartments on the Refuge and correlated densities with amount of upland and bottomland forest in each compartment.

Powell (1998) developed a stochastic dynamic model to investigate the effects of spatially-distributed silvicultural actions on the Refuge wood thrush population. His is apparently the only model that integrates habitat dynamics with a demographic model for wood thrush. His model was spatially-based but not fully spatially-explicit. That is, wood thrush population transitions were accounted for on a forest compartment-level basis, but the spatial arrangement of the compartments was not explicitly recognized in the model. Type of silvicultural treatment applied to compartments (i.e., burned/thinned versus untreated) determined within-season productivity rate, age and sex-specific survival rates, and sex-specific inter-compartment movement rates. A forest transition model calculated between-season overstory and understory responses to treatments.

MODEL-BASED DECISION MAKING IN CONSERVATION MANAGEMENT

Responsible conservation management implies that decision actions should be optimal in some measure, including those actions designed to minimize risk of an undesirable outcome. An ecosystem approach to natural resource management recognizes that resources are dynamic and most often governed by spatial processes, that responses to management are largely uncertain, and that decisions should not be evaluated on a single response measure (Christensen et al. 1996). Although there exist numerous optimization examples that take into account these concerns individually, none appear to comprehensively address all these issues.

MODEL-CERTAIN APPROACHES

By far, most applications of optimization principles in conservation management are in cases where underlying habitat and population models are considered certain. That is, the model's structure and parameterization are regarded as known, even though other forms of stochastic uncertainty (partial observability, environmental uncertainty, and partial controllability) may occur in the model.

Often forest planning for wildlife considerations involves specifying goals and constraints in terms of landscape configurations or abundance of specific habitat types. These metrics serve as surrogates for population responses; they may or may not relate to population viability. Kurttila (2001) reviewed many of these habitat-based approaches. Nevo and Garcia (1996) used habitat suitability indices to develop nonlinear programming models for optimal habitat planning. Bettinger et al. (1997) used a heuristic search algorithm for optimal stand harvest schedules that met minimum cover and foraging goals for Rocky Mountain elk (*Cervus elaphus nelsoni*). Loehle (2000) described another heuristic search algorithm that finds an optimal stand harvesting schedule that leaves a remnant patch of sufficient size for wildlife and with minimum patch edge.

However, a number of recent studies have investigated optimization based on models of population growth and dispersal among habitat patches. Hof et al. (1994) solved a mixed-integer programming problem in which forest harvesting decisions through time on an artificial landscape altered carrying capacity, and thus population growth and dispersal, for two species that favored different forest seral stages. Hof and Raphael (1997) combined simulation with linear optimization to explore alternative arrangements of habitat for the northern spotted owl. They used a population simulation model to estimate population limitation and carrying-capacity functions for owls, and they provided these functions in linearized form to a solver. However, theirs was a static optimization, which they justified by pointing out the semi-permanence of habitat decisions (Hof and Raphael 1997).

Haight (1995) analyzed a stochastic metapopulation model to find forest harvesting policies that maximize financial return while meeting pre-specified probabilistic levels of vertebrate species viability. In his model, amount of timber harvest influenced both the carrying capacity of a patch and degree of dispersal from the patch. In a related problem, Haight and Travis (1997) constructed a stochastic metapopulation model for gray wolves (*Canis lupus*) in the upper Great Lakes region, and they described random search techniques to find the minimum amount of habitat to preserve to assure population viability in one of the patches.

Lubow (1996) modeled two spatially disconnected populations of organisms in which demographic and environmental stochasticity, catastrophic events, and density dependence all regulated population growth. He used stochastic dynamic programming (Bellman 1957, Dreyfus and Law 1977) to select number of individuals to translocate between populations, conditional on current size of each population, to maximize probability of long-term persistence of the populations.

Doherty et al. (1999) used dynamic programming to find woodlot management decisions that were optimal for a ratio of bird population size to revenue loss, given current woodlot size distribution within the landscape. They captured output from a stochastic, individually-based, spatially-explicit population model of Carolina chickadees (*Poecile carolinensis*) and used these quantities as single-stage payoff values in the dynamic programming algorithm.

Conroy and Moore (2001) modeled a simple forest system that existed in two seral stages, where transitions between stages were governed by natural succession or were manipulated by harvest. Each seral stage served as source habitat for one bird species, and they embedded simple stochastic source-sink models into a dynamic programming algorithm to derive optimal harvest decisions conditional on the current habitat distribution of the forest and species abundances. Tuck and Possingham (2000) used dynamic programming to analyze harvest of a fishery that comprised both exploited and protected populations in a demographically connected metapopulation.

Moore et al. (2000) modeled forest growth on an artificial landscape divided into management compartments. They used a model of basal area yield to project habitat conditions forward through time and in response to compartment-specific thinning and harvesting decisions. Population dynamics of a generic bird species favoring old-growth habitat were regulated by source-sink relationships and distance-moderated inter-compartmental dispersals. They used a genetic algorithm (Goldberg 1989) to find near-optimal spatial and temporal schedules of silvicultural activities designed to maximize long-term bird abundance, given an initial state of the forest landscape.

Several of the above studies explicitly addressed the issue of multiple objectives and trade-offs between satisfying each objective. In some cases, individual resource objectives were weighted and combined in some fashion to form a single composite objective statement (Hof et al. 1994, Nevo and Garcia 1996, Doherty et al. 1999, Conroy and Moore 2001). In other cases, the optimization procedure explicitly acknowledged one of the resource objectives and constrained solutions to meet threshold values of other objectives (Haight 1995, Nevo and Garcia 1996, Haight and Travis 1997, Moore et al. 2000). Still others used graphical techniques to examine the production possibilities frontier and analyze trade-offs among competing resource objectives (Rohweder et al. 2000). For example, Lin and Buongiorno (1998) optimized a Markovian forest landscape model and graphically portrayed trade-offs in optimal management for maximum forest income, forest diversity, and landscape diversity.

OPTIMAL MANAGEMENT UNDER MODEL UNCERTAINTY

In contrast to the model-certain approaches, model-uncertain, or adaptive, approaches do not presume knowledge of any single model structure or parameterization. Assuming structural knowledge in an uncertain environment may easily lead to sub-optimal (i.e., overly costly either in terms of the resource or in implementation) conservation strategies (Dakins 1999). Adaptive approaches embrace the notion that improvements in management can be gained whenever information is available and applied to the reduction of uncertainty. This is the idea of dual control: some sacrifice in short-term resource objectives can be tolerated if, as an outcome of decision making, knowledge is gained that will lead to better management over the longer term (Walters and Hilborn 1978, Walters 1986:257, Williams et al. 1996). More than 100 years ago, Chamberlin (1890) appealed to the scientific community to employ not one but a family of working hypotheses for the conduct of scientific inquiry. Adaptive management gives Chamberlin's (1890) proposal operational footing and provides resource managers a way out of seemingly intractable and contentious decision problems.

Hughell's (1996) work on forest management for the red-cockaded woodpecker was purportedly an implementation of adaptive management. He used a genetic algorithm to find an optimal spatially-explicit timber harvest schedule given current forest status and woodpecker distribution. He then simulated the schedule for one time period and recorded the expected response by the woodpecker population. With change in the woodpecker distribution likely, subsequent steps of the decision schedule could then become infeasible. Therefore, these steps of optimization and simulation were repeated throughout the planning horizon. This example does not qualify as adaptive management, in the sense of Walters (1986) or Williams (1996), as the model structure and parameters are completely determined. The chance redistribution of birds is apparently the "information feedback" in this problem, and the fact that the decision schedule must change in light of the new system state is evidently the "adaptation" being described. This case exemplifies a misconception in conservation management that the ability to alter decision actions in response to the changing state of a resource is a sufficient condition to classify a management program as "adaptive." In fact, this condition is necessary but not sufficient in an adaptive policy; the sufficiency is satisfied when information is used to update parameters of the resource model.

Anderson (1975) provided one of the earliest treatments of optimal harvesting of a wildlife resource in a stochastic, dynamic environment. His effort is noteworthy because he specifically addressed the issue of uncertainty about the degree of compensation in mallard (*Anas platyrhynchos*) harvest mortality, an issue that continues to vex harvest management decision making (Nichols et al. 1984). Anderson (1975) did not espouse one model over another, but he recommended that management under either model should be periodically evaluated and perhaps adjusted if new data shed further light on model appropriateness. Johnson et al. (1993) later formalized the treatment of model uncertainty when they proposed the development of an adaptive waterfowl harvest strategy.

As stated earlier, adaptive management applications can be described in terms of structural uncertainty type (model structure versus parametric uncertainty) and in aggressiveness of information pursuit (passive versus active). Parametric uncertainty is often addressed through formal Bayesian methods. McAllister and Kirkwood (1998) used a Bayesian analysis to update probability distributions on the parameters of a logistic growth model for a fish stock. Given the growth model, an optimal harvest decision action was identified on the basis of the posterior distribution of a computed catch statistic from the model. Following each decision, data from the fishery were used to again update the model. Pascual and Hilborn (1995) conducted a simulation exercise in which the optimal rate of wildebeest (*Connochaetes taurinus*) harvest was dependent on identification of parameters in a stochastic habitat and population growth model. They estimated posterior probabilities of the parameters via a Bayesian analysis, and they used simulation to determine how well alternative recruitment hypotheses (different specific settings of parameter values) were distinguished under each harvest regime. In both of these cases, information was pursued passively as the different harvest options were not evaluated in terms of future information gain (Collie and Walters 1993).

A weakness exhibited in the above Bayesian approaches to optimal decision making is that, although system processes are dynamic, management actions, once chosen, are assumed to be fixed through time. This is not realistic management behavior for a large class of dynamic problems, and Collie and Walters (1993) pointed out the difficulty in simulating expected management actions in response to future system behavior and information accrual. Dynamic programming methods, on the other hand, entertain the possibility of different optimal management actions with changing future system states over finite or infinite time horizons (Bellman 1957, Dreyfus and Law 1977). In addition, transitions in the information state of the system may be modeled so that optimal actions could also depend on the current degree of knowledge about the system (Williams 1996). These methods, however, require discrete representation of the entire system, that is, all state variables, stochastic variables, and decision variables must take on discrete values (Lubow 1995). Therefore, dynamic optimization approaches that address model uncertainty do so with respect either to different model structures or to different distinct values of parameters.

Johnson et al. (1997) used dynamic programming to optimize a model of waterfowl harvest under uncertainty about model structure. They constructed a population dynamics and harvest model for the continental breeding population of mallard. They proposed two alternative hypotheses concerning degree of density dependence in reproduction and two hypotheses regarding degree of harvest compensation, thus yielding four alternative models of population dynamics. Each model was optimized individually, and each produced a state-specific table of harvest decisions that maximized a total harvest function over an infinite time horizon. A composite table was also generated, in which a weighted sum of harvest returns by all four models was optimized. Equal belief weights of 0.25 were assigned to each model, representing complete uncertainty among models. The optimal harvest strategy thus produced was passive, in that changes in the information state were not anticipated in the optimization. However, Williams (1996) developed a computer algorithm to derive active adaptive decision policies dependent on both resource state and information state. He demonstrated his program on the models developed by Anderson (1975).

An alternative approach to representing system uncertainty is through a Bayesian belief network (Rieman et al. 2001). Here, one builds a single network that qualitatively describes process pathways of the managed system and assigns conditional transition probabilities between network nodes. The network is then analyzed to estimate probability of occurrence of specific outcomes, conditional on the network probabilities. Data collected from the system can be used to update the estimates of conditional probability. The advantage of using such networks is that conditional probabilities can be established either empirically, when data are available, or subjectively based on expert opinion when data are lacking (Rieman et al. 2001).

RESEARCH JUSTIFICATION

Previous authors have urged adaptive approaches to conservation management (Irwin and Wigley 1993, Nichols et al. 1995, Lancia et al. 1996, Marzluff et al. 2000), including recovery efforts for the red-cockaded woodpecker (U.S. Fish and Wildlife Service 2000). Adaptive approaches in natural resource management offer a number of advantages (Williams 1997, Johnson and Williams 1999), particularly in the area of conservation management where scientific uncertainty is profound. Foremost, adaptive management provides greater objectivity and transparency in decision making, in that optimal decisions can be made not only in the face of uncertainty, but in a fashion to actively reduce that uncertainty. Adaptive management forces a clear, public discussion of scientific uncertainty apart from discussions concerning conservation objectives. Thus, scientific uncertainty under this approach cannot be used to cloak fundamental disagreements over management objectives and preferred decision alternatives. Also, adaptive management provides explicit, critical, and collaborative roles for efforts in monitoring, research, management, and policy making.

Unfortunately, examples of adaptive management are rare in conservation management. My research attempts to illustrate adaptive management in a context that integrates spatial modeling, model uncertainty, multi-species response, and monitoring. The application specifically addresses management of the red-cockaded woodpecker and associated songbirds, but the approach could be tailored to address other conservation problems. Uncertainty in this modeled system focuses on the form of the relationships between habitat and woodpecker productivity and between habitat and wood thrush abundance. Decision options are the possible ordering of forest compartments to treat through time and the frequency of prescribed burning.

The models I report here are based on my best understanding of bird ecology and interpretation of limited data. They offer no particular improvements or novel alternatives to models previously developed, and their many weaknesses could be pointed out. However, my intent is to illustrate that perfect models are not needed in the support of decision making, only a set of plausible, if imperfect, alternative models. Model improvements will come as systems are monitored over time and the effects of management on bird populations are better understood. In the meantime, decisions may nevertheless be made even when uncertainty about managed systems is very high. In the sections that follow, I describe methods of data collection, map creation, development, simulation and analysis of models, and optimization analysis.

CHAPTER 3

DATA COLLECTION AND ANALYSIS

This research focuses on management conducted at the 14,136-ha Piedmont National Wildlife Refuge in central Georgia, USA. In the sections that follow, I describe the history and characteristics of the Refuge, data collection efforts, and data analysis.

PIEDMONT NATIONAL WILDLIFE REFUGE

The Refuge is located in Jasper and Jones counties, at the southern edge of the Piedmont physiographic province (Figure 5). The topographic, edaphic, and vegetation features of the Refuge are representative of the region (Lennartz and Heckel 1987, Loeb et al. 1992).

HISTORY

Prior to European settlement in the early 19th century, a hardwood-dominated climax forest likely covered uplands of this region (Nelson 1957). Following settlement and continuing until the Depression, repeated cycles of land clearing, crop cultivation, and farm abandonment occurred throughout the southern Piedmont (Brender 1974). Soil-depleting farming practices destroyed the topsoil and led to widespread erosion in all areas of the region (Brender 1974), including those lands that eventually formed the Refuge. The region quickly reforested in loblolly and shortleaf (*P. echinata*) pine, concurrent with the growth of a vigorous forest products industry in the early 20th century. Natural fire regimes were suppressed in the nascent forest, and aggressive encroachment by shade-tolerant understory hardwood species soon followed (Brender 1974).

Lands acquired under the Resettlement Administration during the New Deal were transferred to the U.S. Bureau of Biological Survey in 1939 for the creation of the Refuge (Gabrielson 1943:109-110). The Refuge was established for the promotion and representation of native upland game wildlife. Very little vegetative cover occurred on the Refuge at the time, and the land suffered severe erosion damage (Gabrielson 1943). The aim of the Refuge was to demonstrate the application of sound management for restoring the degraded landscape and wildlife populations of the region (Czuhai and Cushwa 1968), but because of the extreme degree of damage, Gabrielson (1943) forecast that this goal would not be realized for many years.

DESCRIPTION

The Refuge lies to the east of the Ocmulgee River and entirely within the Ocmulgee watershed. The Hitchiti Experimental Forest (HEF) of the Oconee National Forest (ONF) and some private lands split (approx. 2 km separation) the Refuge into northern (12,127 ha) and southern (2009 ha) tracts. The northern Refuge border adjoins the ONF and some private properties; all other borders adjoin private land. The Refuge envelops 332 ha of private inholdings; most (98%) of this area occurs in four large (40-147 ha) inholdings on the eastern side of the northern tract.

Numerous stream beds and bottomlands dissect the gently to somewhat steeply undulating terrain. Clayey, acidic, and highly erodible Davidson soils occur in most of the upland areas, whereas the more sandy Congaree and Toccoa soils occur in bottomlands (Long and Carr 1916, Payne 1976).

Forests covered most (96%) of the Refuge in 1982 (Refuge Habitat Management Plan). Pine cover, primarily loblolly but also some shortleaf, was the dominant overstory cover type in 75% of the forest. White oak (*Quercus alba*), southern red oak (*Q. falcata*), and hickories (*Carya* spp.) were the dominant overstory species on upland hardwood sites, comprising 12% of the forest. Sweetgum (*Liquidambar styraciflua*) and yellow poplar (*Liriodendron tulipifera*) were the dominant overstory species in bottomland forests. Roads, impoundments, and permanent openings comprised the non-forested portions of the Refuge (Refuge Habitat Management Plan).

Management

The Refuge is divided into 34 management compartments along road and stream boundaries. Compartments are managed to be representative of overall forest composition and age structure of the Refuge (Refuge Habitat Management Plan). Each compartment belongs to one of eight permanent management groups (Figure 6). Assignment of compartments to groups facilitates forest planning on an eight-year compartment visitation schedule. All four (or five) compartments within a management group are evaluated and treated simultaneously, and different groups are treated in different years. Because no two compartments assigned to the same group share a common border, forest treatments are dispersed across the Refuge in any year. Refuge personnel conduct a timber cruise and prepare a management prescription for each compartment in a group. The prescription identifies regeneration and thinning harvest actions on a stand-level basis within the compartment. Private contractors then conduct any forest harvesting actions called for in the prescription.

The silvicultural system used for pine management at the Refuge is retention (irregular) shelterwood (Refuge Habitat Management Plan; J. D. Metteauer, Piedmont National Wildlife Refuge, personal communication), an even-aged regeneration system in which some of the standing shelter trees are left on-site until the following rotation (Smith 1962). In addition to protecting the developing seedling crop, the retained trees may serve as red-cockaded woodpecker foraging (U.S. Fish and Wildlife Service 1998) or nesting (Conner et al. 1991) habitat. Refuge managers designate pine stands as either sapling (P1; <16 yr), poletimber (P2; approx. 16-40 yr), or sawtimber (P3; ≥40 yr).

Pine stands are managed on an 80-year average rotation length; the specific length depends upon site fertility (Refuge Habitat Management Plan). A recovery plan for the red-cockaded woodpecker, however, calls for a minimum rotation length of 100 years for loblolly pine (U.S. Fish and Wildlife Service 1998). A series of intermediate thinnings is carried out over the rotation to achieve specified density and crown closure goals for red-cockaded woodpecker (Refuge Habitat Management Plan). Degree of crown closure is used to further classify sawtimber stands: dense (P3A; ≥70% closure), intermediate (P3B; 40-70%), and sparse (P3C; <40%) closure.

Refuge managers used dormant-season fire sporadically prior to the 1960s (Czuhai and Cushwa 1968) but more consistently since (Lennartz and Heckel 1987; J. D. Metteauer, Piedmont National Wildlife Refuge, personal communication). Managers evaluate need for fire annually on a Refuge-wide basis. Fire use through the 1990s was concentrated along the west side of the Refuge (Figure 7) coinciding with the distribution of red-cockaded woodpecker clusters. Managers attempt to burn woodpecker nesting and foraging areas on a 2-3 year cycle, but inter-burning periods often vary among specific sites (J. D. Metteauer, Piedmont National Wildlife Refuge, personal communication).

Hardwood species are aggressively encroaching into pine stands throughout the Refuge (Refuge Habitat Management Plan). Refuge managers use fire and mechanical removal to control the hardwood understory in those stands where woodpeckers exist or are desired. However, in an effort to increase Refuge habitat diversity, managers do not try to impede hardwood succession on many other upland areas (Refuge Habitat Management Plan). Managers forecasted in 1982 that succession will continue until 40% of the forest cover comprises upland (UH) and bottomland (BH) hardwood types (Refuge Habitat Management Plan).

DATA COLLECTION

Data required for this research came from several sources. Refuge personnel provided data on red-cockaded woodpecker productivity, general stand conditions, and treatment history. I obtained spatial data for GIS development from public spatial data sources. My University of Georgia research colleagues (M. C. Anderson, L. F. Dawood, W. T. Plummer) collected data on bird and vegetation status in conjunction with this study. These data provided estimated wood thrush densities (W. T. Plummer, University of Georgia, unpublished data) which I used in my management models.

WOOD THRUSH SURVEYS

My colleagues collected counts of wood thrushes during 1996-2000 in transect-based distance surveys of forest birds; complete details of methods and results regarding these surveys are described elsewhere (M. J. Conroy et al., Final Report: Adaptive Management Framework for the Monitoring of Forest Wildlife Populations on National

Wildlife Refuges, 2001, unpublished report) (hereafter, “Final Report”). A survey route was established in each of ten Refuge compartments (compartments 5, 6, 8, 11, 12, 16, 23, 24, 25, and 31) in 1996 and in two additional compartments (27 and 33) in 1997. Survey compartments were chosen to represent the range of treatment actions (no action; thinning or burning only; combined thinning and burning) expected to occur throughout the Refuge over the course of the study.

Survey routes often followed Refuge logging roads, streams, county gravel roads, or Powell’s (1998) wood thrush transect lines, but many followed no pre-existing course. Route length varied (2.4-4.8 km) depending on the compartment size, shape, and availability of usable paths. Routes were laid out in closed circuits to minimize multiple encounters of individual birds, and they were configured to sample habitats in proportion to their occurrence in the compartment.

Under relatively calm conditions (no rain or high wind), a single observer walked the survey route during a three-hour period starting at daybreak. For each detected wood thrush, the observer recorded time of day, overstory habitat (by direct sight, or if distant, by judgment from Refuge stand maps), detection type (visual or audible), and estimated distance to the bird. The observer recorded each detection without regard as to whether the bird was detected alone or in a pair.

Transects were surveyed seven times each breeding season, except in 1997 (five times each) and in 2000 (three times each). The reduction of survey effort in 2000 coincided with the start of a Refuge-wide point count survey (Final Report). Counts were conducted in May and June of each year, except in 1996 when some surveys continued into July.

Survey protocols were adjusted over time in response to problems encountered in the surveys. Routes were redesigned after 1996 to better reflect compartment habitat compositions. Also that year, the practice of obtaining distance by extrapolating a measurement between subjectively placed points on a map was abandoned in favor of directly estimating distance in the field. Prior to 1999, all surveys of a single route were usually assigned to a single observer. To diminish the effect of within-year observer variability among routes, observers since 1998 were rotated among the different surveys for a single route. Though many standardization protocols (e.g., selection of consistent timing and environmental conditions) helped provide among-year survey consistency, observer detection proficiency nevertheless changed each year with the composition of the survey crew (Final Report).

VEGETATION SURVEYS

Surveys of vegetation conditions along bird transects were carried out in July and August of 1996-1999. On each transect, the survey team paced 100-m intervals and established a series of survey stations on alternating sides of the transect, 50 m perpendicular from the transect. In 2000, vegetation data were collected in conjunction with the songbird point count survey (Final Report) at 240 survey stations distributed across 21 Refuge compartments. For the 2000 survey, each survey station comprised five sampling points, with four points arrayed in each of the cardinal directions 50 m distant from a central sampling point.

Attributes of vegetation and physiography were measured at each station or survey point, following modifications on methods suggested by James and Shugart (1970) and Noon (1981). Overstory attributes included dominant overstory type (BH, UH, P1, P2, P3, or open), secondary overstory type, if any, and degree of canopy closure (0-100%). Total basal area (m^2/ha) was measured for both pine and hardwood components. Vertical vegetation density at each of three height strata (0 m, 0-2 m, >2 m) was estimated as the proportion of ground obscured by vegetation in a 10-m radius circle. Horizontal vegetation density was estimated as the proportion of squares on a density grid board, 10 m away from the observer, that was not obscured by vegetation. Measurements were taken in three height strata of the density board: 0-0.3 m, 0.3-1.0 m, and 1.0-2.0 m. Slope position (top, bottom, midslope) and slope aspect were also recorded at the site.

COLLECTION AND ASSEMBLY OF SPATIAL DATA

I created a GIS to supply data in support of spatially-based forest and bird modeling efforts for the Refuge. The principal source of data for the GIS was the set of compartment type maps maintained by the Refuge (J. D. Metteauer, Piedmont National Wildlife Refuge, personal communication). A type map reflects general land cover conditions in a compartment as well as physical features including roads, trails, stream beds, and structures. Refuge forest managers revise the type map every eight years when management prescriptions are being prepared for the compartment. Thus, a complete set of the most recent compartment maps does not reflect Refuge-wide conditions at a single point in time. My goals were to

1) Digitize the most recent type maps to create a complete digital spatial record of Refuge features and land coverages,
2) Estimate physiographic attributes of the landscape from digital elevation models,

3) Render the GIS into a cellular representation suitable for forest and bird population modeling, and
4) Synchronize elements of the GIS to reflect estimated contemporaneous forest conditions at any chosen point in time.

POLYGON AND LINEAR FEATURES

Refuge personnel scanned the most recent version of each of the 34 Refuge compartment type maps into a JPG-format image file. I used digital aerial photographic images (U.S. Geological Survey 3.75-minute digital orthophoto quarter quadrangle, 1993 photograph date) to register Universal Transverse Mercator (UTM) coordinates of prominent landscape features on the maps. I then projected the map images into a UTM coordinate system using the ARC/INFO (ESRI, Inc.) Rectify command.

I created several theme layers from the type map images. I first created a compartment theme (Figure 8) by digitizing compartment boundaries directly from the type map images. I digitized compartments in spatial sequence starting in the northeast corner of the Refuge (compartment 5). Each new compartment was digitized by appending a polygon to the set of polygons already completed. Because type maps had been created in isolation of one another, and because the digitizing work proceeded from northeast to southwest, compartment boundaries in the GIS and in the type maps tended to disagree with each other mostly along the northern and eastern sides. These discrepancies were most apparent where upper reaches of streams separate compartments (e.g., 5 and 12, 2 and 6), as these stream channels are often hard to identify on a photograph and transfer to a type map. I excluded private inholdings on the Refuge from the compartment theme.

I split the compartment polygons into individual forest stands and non-forest land types to create a stands theme (Figure 8). In accordance with the type maps, I assigned each stand polygon to one of the nine type classes depicted in the Refuge Habitat Management Plan: pine types P1, P2, P3A, P3B, and P3C; hardwood types UH and BH; open land; and impoundments. Based on descriptions of general silvicultural conditions at the time of the most recent management prescription, I assigned approximate age and basal area (m^2/ha) map attributes to all P2 and P3 pine stands in the compartment. These measures were often no more than midpoints of roughly-estimated data ranges, and I applied a single set of age and density measures to all stands of the same type class within a compartment. In cases where management prescription reports supplied no estimate at all of pine type age and basal area, I used approximate values based on size-age relationships summarized in the Refuge Habitat Management Plan. These age and basal area approximations were 28 years and 16.1 m^2/ha (70 ft^2/ac) for P2, 64 years and 19.5 m^2/ha (85 ft^2/ac) for P3A, 64 years and 16.1 m^2/ha for P3B, and 64 years and 12.6 m^2/ha (55 ft^2/ac) for P3C. In contrast, type maps usually supplied a distinct year of establishment for each P1 stand, and I input these values to the GIS. Basal areas of P1 stands were unknown, but I later estimated P1 basal area using a variation of the Bailey and Ware (1983) mensuration model (see *Forest State Synchronization*, below).

I converted a copy of the stands polygon theme into a line segment theme. From this theme, I created separate themes for roads and streams (Figure 8) by joining or cutting away certain line segments. Therefore, roads and streams that form stand or compartment boundaries were portrayed consistently with those boundaries. I gave road and stream segments unique identifiers according to a “tree-and-branch” nomenclature system. For example, the identifier A.300.140.030 in the streams theme represented the 3rd tertiary branch of the 14th secondary branch of the 30th primary branch (Allison Creek) of the main stem A (Falling Creek).

PHYSIOGRAPHIC GRID THEMES

I calculated two grid layers, slope position and site index, from a 30-m resolution digital elevation model (DEM) (U.S. Geological Survey 7.5-minute DEM, published 1979). First, I used the Spatial Analyst tool in ArcView GIS (ESRI, Inc.) to calculate slope (G_i'; i.e., first elevation gradient), curvature (G_i''; second elevation gradient), and aspect (Q_i) values for each pixel i in the DEM. I compared the elevation of each pixel to the range of elevations in a 13×13 square neighborhood of pixel i, and I classified the focal pixel as to whether its elevation occurred in the upper ($U_i = 1$) or lower ($U_i = 0$) half of the neighborhood elevation range. For each pixel i, I created a categorical attribute, slope position P_i, that took on one of five ordinal values according to the following rules:

P_i	Condition
1: Bottom	$(G_i' \leq 5\%)$ and $(G_i'' \geq 0.1$ or $U_i = 0)$
2: Foot	$(G_i' > 5\%)$ and $(G_i'' > 0.2)$
3: Mid	$(G_i' > 5\%)$ and $(-0.2 \leq G_i'' \leq 0.2)$
4: Shoulder	$(G_i' > 5\%)$ and $(G_i'' < -0.2)$
5: Top	$(G_i' \leq 5\%)$ and $(G_i'' \leq -0.1$ or $U_i = 1)$ and $(P_i \neq 1)$

I predicted site index for each pixel as a function of slope (% slope/100), aspect (degrees azimuth) and slope position:

$$S_i = \exp[\, b_0 + b_1 G_i'\cos(Q_i) + b_2 G_i'\sin(Q_i) + b_3 G_i' + b_4(P_i\text{-}1) + \epsilon_i(\sigma^2)\,],$$

where S_i is predicted site index (m, at base age 50 years) for pixel i and $\epsilon_i(\sigma^2)$ is a random variable drawn from a normal distribution with mean 0 and variance σ^2. The values b_k and σ^2 are model parameters. This function was based on one by Stage (1976), who proposed the use of slope-aspect interactions and additive habitat effects in predictive models for site quality. I used a stochastic version of the model, however, and I included a parameter, b_4, that related site index to slope position, here expressed as an ordinal value. At the Refuge, slope positions are approximate indicators of forest ecotypes or habitats, among which I would expect site index to vary.

I chose parameter values for this function subjectively through trial and error. However, I attempted to calibrate the model so that model output matched historical summary estimates of site index. Measurements based on 1% cruise samples carried out across the Refuge in 1982 yielded a mean site index value of 24.8 m (81.3 ft) for upland pine stands and 30.2 m (99 ft) for bottomland hardwood stands (Refuge Habitat Management Plan). A study on the neighboring HEF provided a coefficient of variation value of 12.8% for site index (Bailey and Ware 1983). For each trial set of parameter values, I used the model to calculate a site index value for every pixel. I randomly assigned the pixels into 100 equal-sized groups and calculated summaries of predicted site index values within each group. I did so to replicate the 1982 sampling procedure and to thus provide a basis of comparison of the model output to the historical estimates. Following many such trials, I settled on the values $b_0 = 3.407$, $b_1 = 0.8$, $b_2 = 0.8$, $b_3 = -0.13$, $b_4 = -0.074$, and $\sigma^2 = 0.103^2$. In most of the 100 sample groups, these parameter values yielded site index summaries within three decimal places of the historical means and variances for both measured habitat classes. Based on predictions from this model (Figure 9), the approximate range of site index values for the entire Refuge was 14.7-46.6 m (48.3-152.8 ft), and the approximate central 90% of the distribution ranged 20.4-33.6 m (66.8-110.1 ft). Site index range for non-bottomland sites ($P_i \geq 4$) was 14.7-40.7 m (48.3-133.7 ft), and central 90% distribution range was 20.8-29.0 m (68.2-95.0 ft). I used SAS (SAS Institute, Inc.) to conduct all analyses of site index (Appendix C.1-C.2).

Cellular Rendering of the GIS

Because modeling and decision support activities are difficult to carry out on a map with irregular polygonal features, I rendered elements of the map in a theme layer of 4.05-ha (10-ac) hexagonal cells (Appendix C.3-C.5). Hexagons have simple geometric properties yet they permit a degree of realism and flexibility in the construction of spatial models. Their use is becoming commonplace in spatially-explicit population models of red-cockaded woodpecker (Hughell 1996) and other birds (Pulliam et al. 1992). I chose the 4.05-ha hexagon size for consistency with management recommendations for minimum red-cockaded woodpecker cluster size (U.S. Fish and Wildlife Service 1998, 2000).

Using ARC/INFO (ESRI, Inc.), I intersected a grid of 4.05-ha hexagons with the compartment theme constructed earlier. To each hexagon, I assigned the identifier of the compartment comprising the greatest area within the hexagon. Hexagons that intersected the Refuge property boundary were split along the boundary, and I calculated the area contained in the portion that remained. I dropped from the resulting hexagon coverage any hexagon fragment smaller than 0.1 ha (0.247 ac), as such fragments contained unreliably small samples of physiographic pixel data (see below). Fifty-one of 3840 hexagons and hexagon fragments (1.3%) fell below this threshold; together they accounted for only 1.5 ha (0.01%) of total Refuge area.

I also intersected the hexagon grid with the stands theme to assign stand attributes to each hexagon. For each of the four non-pine classes (BH, UH, open, water) found within a hexagon, I summed areas of all habitat fragments within class. For each pine class found within a hexagon, I set age and basal area of each fragment to the values for the largest fragment of that class. For example, age and basal area of all P3B stand fragments within a hexagon were set to the values for the largest P3B fragment. For hexagons contained entirely within a compartment, this reassignment of stand age and basal area had no effect on P2 or P3 stands because these stands had already been assigned consistent values for age and basal area. However, P1 stands within a compartment could vary in age, so the reassignment likely altered ages of some P1 fragments. Likewise, fragments of stands within hexagons that crossed compartment boundaries were also affected by the reassignment of age and basal area.

After fragment ages and basal areas were made consistent within each hexagon and pine type class, I calculated total fragment area by pine type class and identified those classes occurring in greatest and second-greatest abundance within the hexagon. Areas contained in other pine classes, if any, were absorbed by the two dominating classes in proportion to the areas occupied by these two classes. Consolidating pine areas into fewer classes per hexagon simplified the data structure and facilitated modeling, and the procedure introduced negligible error in pine class area representation (on average, 99.1% of all pine area in each cell was contained in one or two pine classes). Thus, each hexagon cell contained none, one, or two consolidated pine classes. I denote these consolidated areas as “stands” in further discussion of the GIS and associated decision models. After consolidation of these fragments, I assigned a value of percent canopy closure to each stand based on its type class. I used values of 40% for P3C stands, 70% for P2 and P3B stands, and 100% for P3A stands (Refuge Habitat Management Plan). I used a canopy closure model to calculate the canopy closure value for P1 stands (see *Forest State Synchronization*, below).

Values of management variables were also assigned to each hexagon. From compartment prescription maps displaying the most recent thinning activities, I identified those hexagons that most closely coincided with treatment areas. I assigned a variable representing year of last thinning treatment to each hexagon containing a P2 or P3 stand (Figure 10). I assumed that cells not thinned at the most recent treatment cycle were thinned in the previous cycle, and for those cells, I set year-of-last-thinning to this earlier year value. In a similar way, I assigned values of year-of-last-burning based on Refuge maps of burning activity (Figure 7). However, I analyzed a 10-year burning record (1991-2000) of all compartments to make these determinations. If a cell had not been burned at all in this time period, I set the year-of-last-burning variable equal to 1990, effectively indicating a “never burned” state.

I also intersected the slope position and site index grid-based physiographic themes with the hexagon theme. These intersections assigned grid pixels to hexagons; thus, I obtained a suite of summary statistics of slope position and site index for each hexagon.

Lastly, I calculated a set of feature distance and buffer composition measures for each hexagon. I obtained a stream density measure (m/ha) by intersecting the streams theme with the hexagon theme. I also calculated distance (m) from each hexagon center point to the nearest stream. Around each center point, I calculated land area in Refuge ownership, in ONF ownership, and in private ownership occurring in radial buffers of 400 m, 800 m, and 3220 m.

The complete forest data theme thus contained information on the following attributes in each hexagon (Appendix C.5):

1) Hexagon size and location;
2) Compartment membership;
3) Most recent prescription year, year of last thinning, and fire history;
4) Distribution of area in non-pine types;
5) Distribution of area in up to two dominating pine components;
6) Age, basal area, and canopy closure status of each pine component at the most recent prescription year;
7) Summary measures of slope position and site index;
8) Stream nearest-distance and density measures; and
9) Land ownership distribution in circular neighborhoods.

FOREST STATE SYNCHRONIZATION

Because I assembled the GIS from temporally-distinct type maps (range 1984-1998), the GIS to this point did not reflect contemporaneous forest conditions. Therefore, in each hexagon, I used a basal area growth model and a canopy-closure rate model to advance pine stand conditions forward from compartment-specific starting times to a common point in time (year 2000) (Appendix C.6-C.8).

Letting t_j represent the year of the most recent management prescription for compartment j, I advanced pine basal area of compartment j from year t_j to year $T = 2000$.

I applied the Bailey-Ware (1983) mensuration formula to each pine stand in each cell of compartment *j*:

$$B_{t+w} = B_t^{(A_t / A_{t+w})} \exp[\, \beta_1 (1 - A_t / A_{t+w}) + \beta_2 X_\tau (1/A_{t+w} - 1/A_t) / (A_\tau A_{t+w}) + \beta_3 S (1 - A_t / A_{t+w}) \,].$$

Here, B_t is stand basal area at time t, w represents a time span, A_t is stand age at time t, A_τ is stand age at last thinning, X_τ is a "thinning index" (Bailey and Ware 1983), S is the median value of site index for the focal cell, and β_1, β_2, and β_3 are model parameters. For a given compartment *j*, I used values $t = t_j$ and $w = T - t_j$ in the model. I used a thinning index value of $X_\tau = 0.6$ to reflect the "thinning from below" (Smith 1962, Bailey and Ware 1983) type of stem removal carried out by managers at the Refuge. I used the parameter values $\beta_1 = 2.81706$, $\beta_2 = -11935.2$, and $\beta_3 = 0.043493$, as provided by Bailey and Ware (1983) for natural loblolly pine stands measured in the adjoining HEF. Under this mensuration formula, cell-specific values of site index project growth to year 2000 differently for each stand. Values of stand basal area for the same type class, which I assumed to be consistent throughout the compartment at the time of prescription, may therefore vary within the compartment when projected to year 2000 conditions.

I estimated basal area for P1 stands at time *T* with a modification of the Bailey-Ware (1983) model:

$$B_{t+w} = \begin{cases} B_0^{(A_0 / A_{t+w})} \exp[\, \beta_1 (1 - A_0 / A_{t+w}) + \beta_3 S (1 - A_0 / A_{t+w}) \,], & A_0 < A_{t+w} \\ B_0, & \text{otherwise.} \end{cases}$$

As before, $t = t_j$ and $w = T - t_j$. The new model parameters reflect a constant basal area value B_0 occurring through age A_0. Beyond age A_0, basal area growth followed the Bailey-Ware model. I chose $B_0 = 0.46$ m^2/ha and $A_0 = 8$ years because these values projected basal area for an unthinned stand to reach 24.3 m^2/ha (106 ft^2/ac) by age 51, the average density and age of dense P3A stands as reported in the Refuge Habitat Management Plan.

I projected canopy closure using a growth formula by Valverde and Silvertown (1997):

$$C_{t+w} = 100 - (100 - C_t)\,\gamma^{w}$$

where C_t is canopy closure (%) at time t and γ is a canopy closure rate. As did Powell (1998) in his forest succession model, I set the rate of closure at $\gamma = 0.79$ (Valverde and Silvertown 1997). For P2 or P3 stands in compartment j, C_t was the canopy closure amount at the initial time t_j, and $w = T - t_j$. For P1 stands, $C_t = 0$ and $w = A_T$, that is, canopy closure amount was a direct function of age at time T.

Refuge personnel provided me UTM coordinates of the center of each red-cockaded woodpecker cluster. I assigned each cluster to the center point of the hexagon containing the cluster, and no cell received more than one cluster assignment (Figures 7, 10). I also recorded in the GIS the year 2000 activity status of each cluster. Because each cell could contain at most one cluster, the terms “cluster” and “cell” appear interchangeably throughout the text wherever it is clear that model structure is being discussed.

ANALYSIS OF SURVEY DATA

Population densities of wood thrush were estimated from distance data collected on the bird transects (Final Report; W. T. Plummer, University of Georgia, unpublished data). Estimates were obtained for years 1997-2000 only; inconsistencies in survey procedures in 1996 precluded use of those data.

Distance methods (Buckland et al. 1993) were used to estimate bird densities. Distances from the observer to wood thrushes, detected either aurally or visually, were supplied to program DISTANCE (version 3.5; Thomas et al. 1998) as were transect length and dominant overstory type associated with the detection. For portions of transects following compartment boundaries, portion lengths were halved as birds were surveyed on only one side of the transect. Three treatments of the data improved estimation of detection functions. First, bird detections were pooled over all habitat types and compartments within years. Distances were right-truncated to 120 m or to the 95% distance percentile, whichever was smaller (Buckland et al. 1993). Lastly, because of the high degree of error that likely occurred in the field estimates of distance, the default distance intervals calculated by program DISTANCE were used in the computations (Thomas et al. 1998). The AIC value (Akaike Information Criterion; Akaike 1973, Burnham and Anderson 1998) indicated a superior detection model function among the alternatives (uniform cosine, half-normal hermite polynomial, half-normal cosine, hazard rate cosine, and hazard rate simple polynomial). Given the annual estimated detection function, distance data were post-stratified (Buckland et al. 1993) for density estimation according to habitat type and compartment membership. Thus, wood thrush density estimates were specific to combinations of years, compartments, and overstory (pine versus hardwood-dominant) types.

I used habitat and compartment-level estimates of wood thrush density to develop habitat-association models. Data from vegetation transect surveys served as predictor variables for these models. I summarized these data by averaging vegetation attributes from all sample sites within an overstory type and compartment.

CHAPTER 4

DECISION MODELING

As discussed earlier (Chapter 1), the making of optimal decisions for a dynamic resource has four requisite components: (1) a measurement of system state, (2) an explicit statement of management objectives, (3) a set of alternative management decisions available at each decision opportunity, and (4) a model describing how the system state and management objectives change in response to decision actions and other influences (Figure 2). The system state for the Refuge includes forest (overstory status, age structure, understory conditions) and bird population conditions. The system state is informative of not only the abundance of the various attributes but also their spatial distribution. A more thorough discussion of the Refuge's monitoring efforts appears in a later chapter.

The focus of this chapter is on development of the latter three components above. My interest was in how the activities of forest cutting (regeneration and thinning), prescribed burning, and installation of artificial woodpecker cavities could be distributed through space and time to bring about desired outcomes for both woodpecker and wood thrush populations. Therefore, I built a set of stochastic, spatially-explicit simulation models, each of which forecast the state of the forest landscape, the abundance of birds, and the distribution of birds in annual time steps in response to these activities (Figure 11). The set of alternative models reflected my uncertainty regarding aspects of forest composition dynamics, woodpecker habitat-productivity relationships, and wood thrush habitat-abundance relationships. Models therefore differed with respect to parameters that controlled these particular dynamics. The representation of uncertainty through a set of alternative models is a novel feature in this work and underscores my main argument that the formal recognition of structural uncertainty is prerequisite to adaptive management for conservation objectives.

To capture management activities in a model, I found it necessary to distinguish strategic management decisions, which were made at the Refuge-wide level, from more tactical management actions, which were essentially a fixed set of action rules to be followed at the stand level. For example, given direction on which Refuge compartments to visit for cutting treatments in a particular year, prescriptions of exactly which stands to treat and by which treatment were dictated according to adjacency constraints, stand age priorities, and other fixed criteria. However, the choice of which compartments to visit was a more strategic type of decision and was one that I explored using the models. Given (1) a specific compartment visit schedule through time and (2) a number of compartments to be burned annually, my models executed a fixed set of rules to simulate the thinning and regeneration of stands in certain landscape cells, the installation of artificial nest cavities in other cells, and the selection of compartments for burning (Figure 11).

The forest responded to the variety and distribution of treatments throughout the landscape (Figure 11). Some areas in pine cover converted to hardwood according to a stochastic rate of hardwood succession. Basal area and canopy closure of the residual pine forest increased in untreated areas and decreased where cutting occurred. Stochastic disturbances regenerated patches of forest. Understory vegetation density responded to changes in attributes of the overstory.

Bird populations then responded to the change in the forest state (Figure 11). I used an individual-based, process-oriented, spatially-explicit population model to project the population of active and inactive woodpecker clusters through time and space. Clusters either remained active or were abandoned according to amount of hardwood midstory that occurred in the cluster cell. Similarly, inactive clusters became active if hardwood midstory was not excessive; however, a proximate and sufficiently large pool of dispersing young was also necessary to change the status of an inactive cluster. In contrast, I used a population-level, statistical, non spatially-explicit model to predict the abundance of wood thrush in each compartment. The prediction model used overstory and understory attributes of the pine cover component of the forest.

Thus, the models were hierarchical in nature (Figure 11). Forest treatments, in part, regulated the forest response, and, in turn, the new state of the forest influenced responses by the woodpecker and wood thrush populations.

I used each alternative model to find an optimal treatment combination (compartment visitation schedule through time and number of compartments to burn annually) for each of three population metrics. The metrics were all long-term (100 yr) population outcomes, and they differed in degree of influence provided to red-cockaded woodpecker and wood thrush population objectives.

This chapter is divided into three sections. The second section details the layout of the spatially-explicit simulation model. Within the model, a set of fixed rules prescribes how regeneration cuts are to be distributed throughout the landscape.

However, guidance on the total amount to cut is provided externally to the model via output from a companion model. This model, a non-spatially explicit forest overstory compartment model, is described in the first section of the chapter. I used this model to derive optimal regeneration amounts in each of three pine age classes, given the current cover type composition of the forest, for achieving an old-growth pine habitat objective. The decision table provided by this compartment model was accessed by the spatially-explicit model to determine the regeneration amount at each time step. Thus, by this approach, I separated the question of *how much* to regenerate annually from the issue of *where* to locate the regeneration. The third section of the chapter describes the process I used to construct the list of decision alternatives, the analysis of decision outcomes under each of the models and under uncertainty among models, and the updating of the information state through comparison of model predictions to field observations.

OVERSTORY TRANSITION MODEL

A plan for recovery of the red-cockaded woodpecker (U.S. Fish and Wildlife Service 1998:28-32) provided guidelines for regeneration of pine forest stands. Specifically, the recovery plan called for the annual amount regenerated to be proportional to the ratio of compartment visitation periodicity to rotation length. Given the minimum loblolly rotation length of 100 years (U.S. Fish and Wildlife Service 1998) and given the Refuge's compartment visitation periodicity of eight years, the fixed amount regenerated by this "area control" method is 8% of the pine acreage each year. Recognizing that age distributions in most national wildlife refuge forests are not currently uniform, the recovery plan recommended avoiding harvests of the two oldest age classes (where age classes are defined by compartment visitation periodicity) in any year, until a uniform age distribution is achieved (U.S. Fish and Wildlife Service 1998).

The recovery plan therefore suggested that when age distribution is not uniform, regeneration activities should be targeted at specific age classes. Furthermore, hardwood encroachment on the Refuge is ongoing and is projected to reach a maximum level of 40% of total Refuge area (Refuge Habitat Management Plan). Therefore, regeneration decisions made today that do not take into account future expected losses in pine to hardwood encroachment may be suboptimal in terms of providing sufficient nesting and foraging habitats for the woodpecker over the long term. Finally, annual losses of pine to hardwood recruitment and to environmental disturbances are stochastic events. Treating such events as deterministic rather than stochastic may also be suboptimal.

To determine if optimal regeneration decisions were dependent on current forest age structure, on hardwood encroachment, and on stochastic pine losses, I built and optimized a dynamic model describing transitions among pine age classes and upland hardwood forest types. The model was in the form of a Markovian stage-based matrix model (Caswell 2001). Parameters of the model were rates of transition among these age classes and between the age classes and the upland hardwood component.

I considered four pine age classes in the model, P1 (age 0-16 yr), P2 (16-40 yr), P3 (40-80 yr), and P4 (≥ 80 yr). For modeling purposes, I split the Refuge's single ≥ 40-yr pine class into the classes P3 and P4 because of the need to specifically recognize and manage for very old stands suitable for red-cockaded woodpecker nesting habitat.

I used the model to derive a stationary harvest policy for maximizing the amount of persistent old-growth forest (age class P4) at the Refuge. The policy was indexed by relative amounts of forest in each of the classes P1-P4 and UH. I did not consider the BH class in this model because I assumed that bottomland hardwood habitats remained unchanged over time. However, the upland hardwood cover class was dynamic in the model, allowing the possibility of pine succession to hardwood. Each unique distribution of forest habitat types indicated optimal amounts of forest from classes P2-P4 to annually regenerate. The array of decision values served as a large "look-up" table of regeneration quotas called upon by the Refuge management simulation model (see *Spatially-Explicit Forest Management Model*, below).

MODEL DEVELOPMENT

The age composition within each pine age class was unknown, but I assumed it to be uniform. I also assumed that mortality and harvest decisions equally affected all ages within an age class.

A vector $\boldsymbol{y}_t$ held Refuge-wide proportions of the four pine age classes and the upland hardwood class occurring at time t. The model projected this composition vector from year t to year t+1:

$$\boldsymbol{y}_{t+1} = \boldsymbol{\Lambda}_t \ \boldsymbol{y}_t \ .$$

Here, $\boldsymbol{\Lambda}_t$ is the matrix product $\mathbf{C}_t \ \mathbf{G} \ \mathbf{H}_t \ \mathbf{D}_t$. The components of this product represent processes of forest disturbance ($\mathbf{C}_t$), growth ($\mathbf{G}$), hardwood encroachment ($\mathbf{H}_t$), and regeneration decisions ($\mathbf{D}_t$), respectively. All processes except that of growth (matrix $\mathbf{G}$) were time-specific, and thus are indexed by t. The five forest class proportions (P1, P2, P3, P4, UH) occurred in positions 2-6 of the state vector $\boldsymbol{y}_t$. Position 1 was a forest class used to temporarily store the current year's regeneration cut (see below).

The first step of the model applied a set of cutting decisions $\{d_{2t}, d_{3t}, d_{4t}\}$ to the pine age classes P2, P3, and P4 at time t. These decisions transferred $d_{2t} + d_{3t} + d_{4t}$ amount of pine forest into a temporary "regeneration" class (Figure 12). This regeneration class was used only for accounting purposes in the model and prevents newly regenerated forest from immediately growing into the P2 class. After the model calculated the growth transitions (**G**, see below), the regeneration class was completely emptied into the P1 class and was thus zeroed out. Matrix $\mathbf{D}_t$ was as follows:

$$\mathbf{D}_t = \begin{bmatrix} 1 & 0 & d_{2t} & d_{3t} & d_{4t} & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1-d_{2t} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1-d_{3t} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1-d_{4t} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

The model next calculated portions of the pine forest lost to hardwood encroachment (Figure 12). I assumed that the rate of loss is persistent each year but is controllable by the amount of harvest taken from each pine class. Thus, through regular cutting, managers may slow or stop the rate of pine cover loss. I assumed that the loss rate, e_{it}, $i = 1, 2, 3, 4$ for the 4 pine classes, was stochastic. The realized rate of loss, E_{it}, was the positive amount of e_{it} that exceeded d_{it}, i.e.,

$$E_{it} = \max(\ 0,\ e_{it}\text{-}d_{it}\ ).$$

Note that because $d_{1t} = 0$ (i.e., no harvest in P1), $E_{1t} = e_{1t}$. Each simulation year, I drew a random variate x_t from a beta distribution with mean μ and coefficient of variation v_e. I obtained the individual e_{it} by applying a set of scaling factors $\mu(e_i)$ to x_t. Thus, the e_{it} were random, but they covaried perfectly with each other. The E_{it} were applied in the model as follows:

$$\mathbf{H}_t = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1-E_{1t} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1-E_{2t} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1-E_{3t} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1-E_{4t} & 0 \\ 0 & E_{1t} & E_{2t} & E_{3t} & E_{4t} & 1 \end{bmatrix}.$$

Following loss to hardwood encroachment, portions of each pine class graduated to the next older class (Figure 12). Transition rates were constant through time and were denoted τ_{ij}, where the transition occurs from the younger class j to the older class i. Parameters τ_{21}, τ_{32}, and τ_{43} described the rates of transition from P1 to P2, from P2 to P3, and from P3 to P4, respectively. A portion of P4 automatically regenerated each year, and I denoted this rate by τ_{14}. Because not all sites on the Refuge are of sufficient quality to support the oldest stands of trees (Refuge Habitat Management Plan), only a portion, k, of the P3 class eligible to graduate actually entered the P4 stage. I assumed that the remainder regenerated as P1. Finally, the model emptied the temporary regeneration class into the P1 class. These transitions appear as follows:

$$\mathbf{G} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1-\tau_{21} & 0 & (1-k)\tau_{43} & \tau_{14} & 0 \\ 0 & \tau_{21} & 1-\tau_{32} & 0 & 0 & 0 \\ 0 & 0 & \tau_{32} & 1-\tau_{43} & 0 & 0 \\ 0 & 0 & 0 & k\tau_{43} & 1-\tau_{14} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

The last component of the model applied an annual stochastic catastrophic or disturbance event c_t to the older pine classes and to the UH class (Figure 12). The event converted the affected pine cover into the P1 type. Only a portion, q, of the affected hardwood type regenerated as pine cover. The rest remained as type UH. The c_t were drawn from a beta distribution with mean $\mu(c)$ and variance $\sigma^2(c)$. The matrix **C** carries out these transitions as follows:

$$\mathbf{C}_t = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & c_t & c_t & c_t & c_t q \\ 0 & 0 & 1-c_t & 0 & 0 & 0 \\ 0 & 0 & 0 & 1-c_t & 0 & 0 \\ 0 & 0 & 0 & 0 & 1-c_t & 0 \\ 0 & 0 & 0 & 0 & 0 & 1-c_t q \end{bmatrix}.$$

The matrix product of $\mathbf{C}_t$, **G**, $\mathbf{H}_t$, and $\mathbf{D}_t$ provided $\mathbf{\Lambda}_t$. Elements in $\mathbf{\Lambda}_t$ specified rates of transition among all compartments of the model (Figure 13).

MODEL PARAMETERIZATION

Thirteen parameters controlled this model. Unfortunately, no data exist from the Refuge to estimate any of them. I chose all of the following parameter values subjectively, though some have more empirical support than others.

I fixed values for the age class transitions (τ_{21}, τ_{32}, τ_{43}, τ_{14}) at (1/16, 1/24, 1/40, 1/40). In other words, I assumed that rates of transition among age classes occurred in proportion to the time span of the younger age class. This is a reasonable assumption if ages are uniformly distributed within an age class. I fixed the value of k at 0.5. This value means that only half of the forest type leaving the P3 age class can enter the P4 class. This parameter setting seemed reasonable for the Refuge, as only approximately half of the Refuge is of sufficient fertility to support the oldest stands of pine (Refuge Habitat Management Plan).

I drew random values of c_t from a beta distribution with mean 0.003394 and variance 0.0003572. This distribution provided a 98th percentile value of 0.05 and a 99.8th percentile value of 0.20. In other words, I expected destruction of 5% of the forest during a 50-year disturbance event and 20% destruction during a 500-year event. These probabilities appear consistent with estimates of tornado (Peterson 2000) and hurricane (Hooper and McAdie 1995) strike probabilities for the region. Below the 90th percentile (i.e., 10-year and more common events), practically none (<0.002) of the forest is destroyed. I fixed q at 0.25 to represent the chance that a hardwood stand destroyed by a catastrophic event regenerates as pine cover.

I had little objective guidance in choosing values for the $\mu(e_i)$ and v_e parameters. After much trial and error, I chose the values (0.0006, 0.015, 0.015, 0.03) for the $\mu(e_i)$, $i = 1, 2, 3, 4$, and I fixed the value of v_e at 40%. The v_e value was small enough to provide a somewhat symmetric distribution of beta variates yet large enough to realistically reflect considerable ecological variability. I believed that the rate of hardwood encroachment should be larger in older than in younger stands, thus $\mu(e_i)$ increases with stand age. Simulations of the model with these values and with harvest rates fixed at the values $d_{it} = \mu(e_i)$ projected the UH component to grow to an asymptotic limit of ~50%. Though these values may be the most arbitrary of the entire set, they are consistent with some empirical evidence. The average of the $\mu(e_i)$, weighted by the age class transition rates τ_{ij}, was 0.012, comparable to the 0.0087 annualized rate of hardwood encroachment in natural loblolly stands in Georgia during the period 1961-1972 (Bechtold et al. 1991).

MODEL OPTIMIZATION

The goal of this work was to derive a state-specific, stationary (time-independent) regeneration policy that maximizes the amount of nesting habitat available for the red-cockaded woodpecker. The estimated minimum amount of nesting habitat needed to support the recovery goal of 96 woodpecker groups (U.S. Fish and Wildlife Service 1998, 2000) is 388.5 ha, or 0.03096 of the modeled forest cover, assuming a 4.05-ha minimum cluster size (U.S. Fish and Wildlife Service 1998, 2000) and assuming total forest cover (pine and hardwood) remains constant. For optimization, I placed primary emphasis on provision of the minimum amount of habitat and secondary emphasis on total habitat amount. Therefore, I maximized the objective function

$$J = \begin{cases} 1000 + y_{\mathrm{P4},\, t+1} - Y_{\mathrm{P4}}, & y_{\mathrm{P4},\, t+1} \geq Y_{\mathrm{P4}} \\ 0, & y_{\mathrm{P4},\, t+1} < Y_{\mathrm{P4}} \end{cases}$$

where $y_{\mathrm{P4},\, t+1}$ is the amount of habitat in the P4 class expected at the next time period, and $Y_{\mathrm{P4}} = 0.03096$ is the nesting habitat threshold. This objective function greatly penalized any decision outcome that did not result in the minimum nesting habitat required. The objective function also recognizes total habitat abundance, but only if the minimum is attained.

I used stochastic dynamic programming to search for an optimal decision policy under this model (Dreyfus and Law 1977). I prepared the model for analysis in program ASDP (B. C. Lubow, USGS Colorado Cooperative Fish and Wildlife Research Unit, personal communication), a successor to the program SDP (Lubow 1995).

One difficult aspect of analyzing this model was the fact that ASDP offers no efficient way to handle dependencies or sum constraints among elements of the system state. The state elements in this forest model ($\boldsymbol{y}_t$) must sum to 1.0. Therefore, I used a set of logistic-power transformations in the model dynamics to make a roughly linear association between ASDP's state vector indices (positive integers) and forest state values (proportions). I applied the following transformation to convert a state variable index I into a logit L:

$$L = \left(\frac{I - \frac{Z+1}{2}}{1 - \frac{Z+1}{2}} \right)^{W} \ln \frac{\pi_0}{1 - M\pi_0},$$

where Z is the number of discrete levels of a state variable, M is the number of independent state variables, and π_0 is a "base" proportion value corresponding to the index value $I = 1$. Parameter W may be any positive exponent. Because the ratio in parentheses may be negative and thus invalidate exponentiation by a non-integer value of W, a negative ratio is first made positive, then the exponentiated result is negated to preserve the sign. For a vector of M indices, a set of M logits is thus obtained. I converted the logits to proportion values in the usual way:

$$\begin{aligned} p_j &= \frac{\exp(L_j)}{1 + \sum_{k=1}^{M} \exp(L_k)}, j = 1, \ldots, M \\ p_{M+1} &= 1 - \sum_{k=1}^{M} p_k \end{aligned}.$$

Five forest components comprised my model, and I represented them with $M = 4$ independent state variables each discretized into $Z = 11$ steps. The values $\pi_0 = 0.001$ and $W = 2.5$ provided a nearly linear relationship between the ASDP index values and forest proportion values. I discretized each of the three decision variables (d_2, d_3, d_4) into eleven values over the range 0.0-0.2 in steps of 0.02. I also expressed probability distributions for both hardwood loss and forest disturbance as discrete distributions with five outcomes each (Appendix C.9).

ASDP searched for an optimal pine regeneration policy indexed to relative amounts of the four pine age classes and the upland hardwood class. I forced ASDP to consider at least 20 stage (year) iterations, but I imposed a maximum computing limit of 500 iterations or the iteration at which the decision policy matrix remained unchanged in three successive iterations, whichever occurred first. At this point, I assumed that ASDP arrived at a stationary optimal policy.

SENSITIVITY ANALYSIS

Because of the uncertainty in the selection of parameters for this model, I performed a sensitivity analysis to measure how the model behaved in response to perturbations in the model parameters. In one form of the analysis, I looked at sensitivity of responses of the overstory class proportions to perturbations of each of the 13 parameters in turn. I used numerical methods to calculate elasticity (Caswell 2001) for each composition response y_j with respect to each of the parameters θ_i:

$$\frac{\theta_i}{y_j}\frac{\partial y_j}{\partial \theta_i} \approx \frac{\theta_i}{f_j(\theta_i)}\frac{f_j(\theta_i+\delta)-f_j(\theta_i)}{\delta}$$

Elasticity is approximated by the relative difference in the *j*th function response when a small (10^{-12}) value δ is added to θ_i. The model responses f_j were the average results of 500 replicate runs of the model simulated under the optimal decision policy over a 500-year time span. To investigate whether sensitivity varied over levels of θ_i, I also calculated elasticity values at $0.5\theta_i$ and $2\theta_i$.

In the second form of sensitivity analysis, I investigated behavior of the optimal decision policy in response to perturbations in the parameters. I used ASDP to calculate optimal decision policies under two alternative scenarios to the default model (model

F0): (1) parameter values that projected a lower rate of transition to the P4 type and a higher transition rate to the hardwood type (model F1) relative to model F0 (Appendix C.10), and (2) parameter values that projected a higher rate of transition to the P4 type and a lower transition rate to the hardwood type (model F2) relative to model F0 (Appendix C.11). I chose values for each scenario as follows:

Parameter	Model F0	Model F1	Model F2
τ_{14}	1/40	1/20	1/60
$\mu(e_1)$	0.0006	0.0012	0.0003
$\mu(e_2)$	0.015	0.03	0.0075
$\mu(e_3)$	0.015	0.03	0.0075
$\mu(e_4)$	0.03	0.06	0.015
v_e	40%	80%	20%
k	0.5	0.25	1.0
q	0.25	0.125	0.5
$\mu(c)$	0.003394	0.001707	0.006711
$\sigma^2(c)$	0.0003572	0.0001816	0.0006911

Annual mortality risk (τ_{14}) of the P4 class is increased in model F1 and decreased in model F2. Similarly, the fraction of P3 eligible to graduate to P4 (k) is reduced in model F1 and increased in F2. Disturbance events are less likely in model F1 (5% destruction in 100-year event; 20% destruction in 1000-year event) than in F0, but they are more likely in model F2 (5% destruction in 25-year event; 20% destruction in 250-year event). I did not alter values for parameters τ_{21}, τ_{32}, and τ_{43} among scenarios because I assumed that they consistently reflected transitions between age classes under any model.

I compared the alternative decision policies with respect to their aggressiveness of cutting actions in each of the pine types. I also compared optimal regeneration decisions for the estimated state of the Refuge forest in year 2000.

SPATIALLY-EXPLICIT FOREST MANAGEMENT MODEL

I constructed a spatially-explicit computer model to forecast responses of the forest landscape, and consequently, responses by red-cockaded woodpeckers and wood thrushes, to alternative decision actions at the Refuge (Appendix C.12). I regulated model behavior by setting model parameters to specific values, so that a user could explore any number of alternative system models through appropriate selection of parameter values. I looked specifically at three groups of alternative settings. One group contrasted the influence of quantity of foraging habitat on red-cockaded woodpecker recruitment. Another compared a linear to a nonlinear description of wood thrush response to habitat. The third considered the three alternative scenarios of forest overstory transitions examined in the sensitivity analysis of the forest overstory model (see *Overstory Growth and Harvest Model*, above).

Given a sequence of compartment-level management decisions to be carried out over time (i.e., a schedule of compartments to visit and the extent of burning), the model simulated the annual processes of management activities, forest response, and bird response (Figure 11). At each time step, the model first simulated stand-level activities of regeneration, thinning, burning, and recruitment cluster creation. Next, the model simulated annual change in forest structure, including hardwood encroachment in pine stands, pine basal area growth and canopy closure, forest disturbance, and understory vegetation growth. Finally, in response to changes in forest structure and landscape, the model forecast cluster activity status of red-cockaded woodpeckers and breeding density of wood thrushes. Thus the model projected the complete state of the forest, expressed in the GIS, from one simulation year to the next. Pertinent model output included predicted number and location of woodpecker potential breeding groups (adult male and female in cluster; U.S. Fish and Wildlife Service 2000) and predicted density of wood thrushes.

The model was spatially-explicit. On a cell-by-cell basis, the model manipulated forest attributes that were captured in the cellular rendering phase of the GIS work. All of the stand-level management actions (regeneration, thinning, burning, recruitment cluster establishment) were dependent on current forest or bird attributes of neighboring hexagons. Additionally, dynamics of the red-cockaded woodpecker population were driven by spatially-computed measures of dispersal potential.

A deterministic set of rules drove all of the stand-level management actions, but several of the forest structure and bird population processes were stochastic. The model randomly chose rates of hardwood encroachment and total disturbance area at each time step. It identified cells at random to regenerate through disturbance. Probabilistic outcomes determined abandonment of active woodpecker clusters, production of young, and occupation of inactive clusters.

I wrote the model in the matrix-based language GAUSS (Aptech, Inc.). Despite the software’s speed in processing matrix-based data elements such as the ones I used, each simulation run was extremely computationally laborious. A single 100-year simulation of the model required approximately ten minutes on a 1.0-Ghz processor.

MANAGEMENT ACTIVITIES

In each simulation year, a list of the specific compartments to visit for cutting treatments and the maximum number of compartments to be burned were externally provided to the model. Under this general guidance, the model simulated execution of management activities on the Refuge. Forest regeneration and thinning were carried out within the identified compartments, but prescribed burning and the establishment of recruitment clusters were activities that occurred Refuge-wide. All action rules carried out by the model were determined by current conditions of the forest and woodpecker populations. The rules were based on constraints and guidelines prescribed either by the Refuge Habitat Management Plan or by one of the woodpecker recovery plans (U.S. Fish and Wildlife Service 1998, 2000). The model simulated management actions in the following sequence (Figure 11):

1) Cells were selected for regeneration from the current management group of compartments;
2) Of those not chosen, other cells in the group were identified for thinning;
3) Based on estimated understory vegetation conditions, entire compartments were chosen from the Refuge at large for prescribed burning; and
4) All cells Refuge-wide were assessed for placement of new woodpecker recruitment clusters.

Forest Regeneration

Each action cycle in the model began with placement of patches for regeneration cutting. Given a group of compartments to visit in the current cycle and given a regeneration quota, the procedure initiated cutting in a "seed" cell chosen from the group, increased the regeneration patch by finding qualifying neighboring cells, and closed out the patch to further cutting upon reaching a patch size limit or fulfilling the quota. If the quota was not satisfied, the procedure repeated these steps for a new seed cell found elsewhere in the management group. The procedure stopped either after reaching the regeneration quota or exhausting candidate regeneration cells.

I identified regeneration quotas for each of pine types P2, P3, and P4 as follows. I calculated Refuge-wide area totals of P1, P2, P3, P4, and UH stands for the current time period from the map of forest attributes. Using these totals as look-up values, I consulted the decision table of optimal regeneration quantities computed in the optimization analysis of the forest overstory model (see *Overstory Growth and Harvest Model*, above). The table (Appendix D) provided optimal amounts of forest to regenerate in classes P2, P3, and P4.

For non-zero quotas, the procedure began a search for cells qualifying for regeneration in the group of treatment compartments. Adhering to woodpecker recovery plan guidelines (U.S. Fish and Wildlife Service 2000), I first eliminated from candidacy any cells containing or adjacent to a red-cockaded woodpecker cluster or a P1 stand. Desirable locations for cutting were those that were far from existing woodpecker clusters and that provided a large quantity of forest area to more quickly satisfy any of the quotas. Isolation from woodpecker clusters reduces the chance that the regeneration cut creates a barrier between the cluster and foraging habitat (U.S. Fish and Wildlife Service 1998). Therefore, I used an average of cluster distance and pine type area to rank each of the candidate cells for selection. For each cell, I calculated mean distance to the three nearest woodpecker clusters. I standardized mean distances by dividing each value by the range of mean distance. Similarly, I obtained amount of hexagon area in each of the P2, P3, and P4 forest types, and I standardized these quantities using corresponding range statistics. For each of the area quantities, I computed a weighted geometric mean of the quantity with the distance measurement. In my use of the model, I provided equal weight to the area and distance components.

For the largest of the three regeneration quotas, the procedure located the cell with the greatest mean distance-area score for that quota type. If none of this quota type remained in the candidate list, the program reduced that quota to 0 and turned to the next largest quota type. Otherwise, the cell identified became the seed cell for a regeneration cut. Within the cell, any pine stand (again, "stands" are seral type subunits of the cell) with an unfilled quota was cut, and corresponding regeneration quotas were reduced by these amounts. These amounts also initialized a patch size accumulator which I used to monitor compliance with the 10.1-ha regeneration patch size limitation (U.S. Fish and Wildlife Service 1998, 2000). Simulation of a "cut" in the model amounted to converting a stand's age, basal area, and canopy closure values to 0. Thus, the model simulates a clearcut rather than a shelterwood type of cut that the Refuge employs.

Following cutting of the seed cell, I investigated the group of up to six candidate cells "ringing" (adjacent to) the seed cell. The procedure identified the ring cell containing the greatest quantity of pine type for the largest quota amount. If none could be found for this quota type, the program looked instead for the cell offering the greatest pine type quantity for the next largest quota amount. Any ring cell selected contained either one or two pine stands. The program cut pine stands in order such that the largest quota amount was reduced first. The area of a cut stand was subtracted from the corresponding pine type quota and added to the patch size accumulator. The program allowed a stand cut only if the regeneration quota for that stand exceeded 0 and if the resulting accumulated patch size would not exceed the patch size limit. The program searched each ring cell in this way until the ring group was exhausted. At this point, the program identified all cells containing a regenerated stand and all their adjacent cells.

These cells were removed from the list of candidate cells, and the program searched for another seed cell. The procedure continued in this manner until all regeneration quotas were achieved or the list of candidates was exhausted.

Thinning

The next process in the forest action cycle carried out pine stand thinning in the selected compartments. All stands older than 16 years of age and exceeding 18.37 m^2/ha (80 ft^2/ac) basal area density were thinned to a residual basal area of 13.77 m^2/ha (60 ft^2/ac). However, stands occurring in a circular 49-cell foraging neighborhood (198.5 ha, 794-m average radius) of a red-cockaded woodpecker cluster (active or inactive) were not thinned if the neighborhood supported less than 50.6 ha (125 ac) of foraging-quality habitat (U.S. Fish and Wildlife Service 1998). Pine stands at least 40 years old and between 9.18-18.37 m^2/ha (40-80 ft^2/ac) in basal area density qualified as foraging-quality habitat (U.S. Fish and Wildlife Service 2000). However, for purposes of assessing stands for thinning, I did not enforce the upper density limit in foraging areas as thinning, by definition, removed that limitation.

Circular foraging areas often occurred beyond Refuge boundaries. In such cases, I multiplied land ownership proportions in an 800-m circular buffer around a cluster by foraging habitat provision rates corresponding to ownership types. A provision rate reflected the proportion of an off-Refuge site providing foraging-quality habitat. I did not know the values of these rates, so I arbitrarily fixed them at 0.5 for the ONF/HEF and 0.2 for private lands. For example, the estimated amount of off-Refuge foraging habitat for a cluster at the center of an 800-m circle (201.1 ha) that comprised 40% private land and 10% ONF/HEF land was 201.1 ha $\times (0.40 \times 0.2 + 0.10 \times 0.5) = 26.1$ ha. I used the same approximations to calculate foraging habitat amount for recruitment cluster placement (below) and for woodpecker recruitment modeling (see *Red-cockaded Woodpecker*, below).

Burning

Following thinning, the model carried out prescribed burning in compartments selected from across the Refuge, up to a given maximum number of compartments. Compartments selected for burning were those that ranked highest on a composite score that assessed understory vegetation conditions in woodpecker nesting habitats, foraging habitats, and all other pine habitats. This approach attempts to balance need for burning with respect to all three components of the forest and is consistent with guidelines expressed in the woodpecker draft recovery plan (U.S. Fish and Wildlife Service 2000). For the nesting habitat score, I computed the maximum vegetation density (see *Understory Vegetation Density*, below) among cluster cells in each compartment. Compartments lacking a cluster cell were assigned the minimum score obtained from all other compartments. The model next computed compartment averages of vegetation density in cells belonging to 49-cell (198.5 ha) foraging neighborhoods of woodpecker clusters. As with the nesting habitat score, compartments containing no foraging habitat cells were assigned the minimum value computed from all other compartments. Lastly, the model computed compartment averages of vegetation density in all cells mostly (>50% by area) in pine habitat and not classified as a foraging habitat cell. For each compartment, I averaged the three scores to obtain the compartment composite score. Based on the composite score ranks, the model simulated burning in the *N* top-ranked compartments, where N was the fixed maximum number of compartments to be burned, by updating the age-at-last-burn variable for all pine stands ≥16 years old.

Recruitment Cluster Placement

The model next determined number and placement of woodpecker recruitment clusters according to guidelines in the red-cockaded woodpecker recovery plans (U.S. Fish and Wildlife Service 1998, 2000). Beginning in simulation year 2001 and in any simulation year evenly divisible by 5, the model calculated a recruitment cluster establishment quota as the number of active clusters times 0.10, the recommended 5-year rate of increase in recruitment clusters (U.S. Fish and Wildlife Service 1998, 2000). According to the guidelines, existing recruitment clusters, which I considered as any intact, non-active woodpecker cluster created either naturally or artificially, count against the establishment quota. Therefore, the number of recruitment clusters to establish in any simulation year was

$$K = \text{quota - existing recruitment clusters},$$

or $K = 0$, if the number of existing recruitment clusters exceeded the quota.

The model placed the K clusters one at a time in cells across the landscape. In each placement trial, the program first found a candidate set of cells which passed all of several location and habitat criteria established by the draft recovery plan (U.S. Fish and Wildlife Service 2000) and then selected a superior cell from the candidate set. The candidate set excluded those cells (a) falling on interior or exterior Refuge boundaries, (b) within 125 m (measured to cell center) of a stream, (c) within 402 m (1/4 mi) of an active or inactive cluster cell (measured between cell centers), (d) containing <85% of area (<3.44 ha) in pine habitat ≥40 years old, (e) containing <50% of area (<2.02 ha) in pine habitat ≥60 years old, or (f) farther than 1609 m (1 mi) from an active cluster. If no cells met these conditions, I relaxed condition (f) in 402-m increments and searched again, up to a maximum distance of 3219 m (2 mi). Initially, the model also eliminated from candidacy those cells (g) not surrounded by a circular 49-cell neighborhood containing ≥50.6 ha of foraging-quality habitat or (h) not surrounded by a circular 13-cell neighborhood (52.6 ha, 409-m average radius) containing ≥25.3 ha of foraging-quality habitat, with off-Refuge habitat contributions approximated according to ownership type (see *Thinning*, above). However, I was forced to drop these latter restrictions as they were too computationally strenuous to evaluate.

Assuming that the candidate set was not depleted at this point, the model selected the cell containing the oldest pine stands (evaluated by area-weighted mean age of stands in the cell) as the location of the recruitment cluster, and K was reduced by 1. I updated the GIS database with the location of this new recruitment cluster. The model continued searching for new cluster cells until K was reduced to zero.

FOREST RESPONSE

Following the simulated management actions, the model advanced the forest state from the current time period into the next, in response partly to those actions (Figure 11). A series of growth and disturbance processes took place, commencing with hardwood encroachment.

Hardwood Encroachment

Refuge managers acknowledge that a significant portion of the current pine landscape will be lost to hardwood encroachment over time (Refuge Habitat Management Plan), and I encoded this scenario in the forest overstory optimization model (see

Overstory Growth and Harvest Model, above). In the spatially-explicit simulation model, I also applied stochastic rates of conversion of pine area to hardwood area. I used the same age-specific average annual rates of conversion as I did in the default overstory model (model F0): 0.0006, 0.015, 0.015, and 0.03 for classes P1, P2, P3, and P4, respectively. Given these age-specific means, and based on variances corresponding to a fixed coefficient of variation of 40% (see *Overstory Growth and Harvest Model*, above), I drew a random conversion rate value from a beta distribution for each pine stand in the landscape. The pine stand area was reduced by that proportion, and the lost area was added to the upland hardwood component in the same landscape cell. Because fire and thinning activities should delay encroachment by hardwood, I did not convert any portions of non-P1 stands which had been burned or thinned in the previous five years.

Basal Area Growth and Canopy Closure

The model next advanced age, basal area, and canopy closure values of each stand into the successive time period. I used the Bailey and Ware (1983) and the Valverde and Silvertown (1997) models for basal area growth and canopy closure almost exactly as described earlier (see *Forest State Synchronization*, Chapter 3). The only departure from before was that $w = 1$ for all stands across the Refuge, that is, every stand was advanced by a single year rather than by a compartment-specific time span.

Forest Disturbance

As did the default overstory optimization model F0 (see *Overstory Growth and Harvest Model*, above), the forest landscape model simulated stochastic forest disturbance events. I drew a random value from a beta distribution with mean 0.003394 and variance 0.0003572 to represent the proportion of the Refuge affected by disturbance events in a year. These values corresponded to a 5% forest destruction rate during a 50-year event and a 20% rate of destruction during a 500-year event. I multiplied the beta variate by total Refuge area to obtain the total forest disturbance, and I distributed the disturbance area randomly to landscape cells throughout the Refuge. Both pine stands in an affected cell converted to age 0 pine. One quarter of upland hardwood area in an affected cell also converted to age 0 pine; the remainder persisted as upland hardwood. Any woodpecker cluster occurring in the cell was destroyed. I did not control the spatial dispersion of the disturbance in any way, therefore large-scale, concentrated disturbance events such as windstorms, ice storms, or wildfires were not spatially realistic.

Understory Vegetation Density

I next predicted understory vegetation density (kg dry weight/ha) in all pine stands across the entire Refuge. The prediction model used was one developed by Conroy et al. (1982) for natural loblolly stands in the southern Piedmont of Virginia. Understory vegetation density responded to overstory basal area, degree of canopy closure, and slope position in their model. I modified this model to reflect suppressed understory conditions due to recent application of fire. Masters et al. (1996) compared burned to unburned control stands managed intensively for red-cockaded woodpeckers in west central Arkansas. They provided data on density of understory vegetation over three years post-burn. I fit a simple linear regression model to their data to estimate degree of understory suppression conditional on number of years since burning. I estimated 0.760, 0.866, and 0.973 rates of suppression for years 1, 2, and 3 post-burn, respectively. I multiplied these estimated rates by output from the Conroy et al. (1982) model to produce a fire-influenced prediction of understory vegetation density in each stand. I computed the mean value of understory vegetation density across both stands in each cell, weighting the average by stand area.

ALTERNATIVE FOREST RESPONSE MODELS

As I did for the overstory transition model, I proposed two models of forest response as alternatives to this baseline model (model F0). For these alternatives, I set parameter values corresponding to those used in the sensitivity analysis of the overstory transition model. The first alternative (model F1) proposed that rate of hardwood encroachment was rapid and rate of forest disturbance was low. Here, I doubled values of HWD_P1, HWD_P2, HWD_P3, HWD_P4, and HWD_CV (Table 1; correspond to overstory model parameters $\mu(e_i)$, $i = 1, \ldots, 4$ and v_e, respectively), halved the value of DSTRB_HW (Table 1; corresponds to parameter q in overstory model), and reduced the mean rate (DSTRB_MN = 0.001707, Table 1) and variance (DSTRB_VR = 0.0001816, Table 1) of forest disturbance (correspond to overstory model parameters $\mu(c)$ and $\sigma^2(c)$, respectively). I made the opposite adjustments to create alternative model F2, which simulated a low rate of hardwood encroachment and high rate of forest disturbance. Values of forest disturbance parameters in this model were DSTRB_MN = 0.006711 and DSTRB_VR = 0.0006911. When simulating these alternative models, I used regeneration decision tables computed in the sensitivity analysis of the overstory transition model (see *Overstory Growth and Harvest Model*, above).

AVIAN POPULATION RESPONSE

Once the model projects the landscape into the next time period (now time t), submodels for wood thrush and red-cockaded woodpecker predict responses by these species to the habitat and, in the case of the woodpecker, to the current population distribution (Figure 11).

Red-cockaded Woodpecker

The red-cockaded woodpecker's requirements for a rare, ephemeral, and specific type of habitat, its dispersal behavior, and its social structure imply a unique course of management focused both at a rather fine spatial scale toward individual breeding groups (e.g., placement of recruitment clusters) and at more extensive scales (e.g., silvicultural planning and application of fire) (Walters 1991, U.S. Fish and Wildlife Service 2000). In Walters' (1991) view, the helper class in a population expands and contracts with variations in survival and productivity whereas the number of breeding groups remains mostly unchanged. Therefore, management to increase survival and productivity of individual birds is less effective in recovering populations than is management that encourages creation of breeding groups, namely, through provision of recruitment clusters (Walters 1991).

The most recent recovery plan for the red-cockaded woodpecker identified the breeding group as the population unit by which recovery should be assessed (U.S. Fish and Wildlife Service 2000), therefore, my woodpecker population model projects dynamics of breeding groups of woodpeckers, rather than individuals. Although the growth of a population is undoubtedly dependent on the population's social structure (Heppell et al. 1994) and the spatial distribution of that structure (Letcher et al. 1998), such representations of the population are not useful for this implementation of landscape-level decision making. First, estimates of transitions among age and social classes are unavailable for the Refuge, and Refuge managers are unable to annually estimate the population structure as birds are not individually banded. Second, it is not clear whether modeling at the level of individuals is helpful or even necessary in a model such as mine in which landscape patterns are highly variable and highly influential in the dynamics of breeding groups.

I developed models that predicted current-year red-cockaded woodpecker cluster activity conditional on prior-year occupancy status and current-year habitat condition and productivity level. I created two complementary models: one which modeled the probability that an occupied cluster would remain active at the next time period (herein referred to as the "persistence" model), and another which modeled the probability that an unoccupied cluster would become active (the "settlement" model).

The question of persistence of cluster occupation is not the question addressed by many cluster occupancy studies. These are often cross-sectional studies comparing habitat attributes of occupied to unoccupied sites (e.g., Kalisz and Boettcher 1991, Loeb et al. 1992, Thomlinson 1995) and thus cannot address occupation probability given previous occupancy status. Walters (1991) and Jackson (1994) noted that red-cockaded woodpeckers tenaciously maintain cluster occupancy, even as surrounding habitat becomes degraded. Therefore, it seems reasonable to presume that a cluster already occupied is more likely to remain in that status than would an unoccupied cluster to become newly occupied in similar habitat.

To calculate persistence probability for occupied clusters, I began with the habitat model empirically derived by Loeb et al. (1992):

$$\Pr(X_{it} = 1 \mid MBA_{it}) = \text{logit}^{-1}(5.134 - 0.4574\, MBA_{it}),$$

where X_{it} is a binary indicator of cluster occupancy in cell i at time t, MBA_{it} is total midstory basal area (m^2/ha) at the cluster site at time t, and $\text{logit}^{-1}(x) = \exp(x)[1+\exp(x)]^{-1}$ is the inverse logit function of x. The form of the occupancy probability curve is sigmoidal, falling with greater values of *MBA* (Figure 14). However, I wanted to postpone the decay of the curve to accommodate a degree of tenacity by cluster occupants. I did so through a linear function that dampened the size of MBA_{it}:

$$MBA_{it}' = 1.2\ MBA_{it} - 8.264.$$

By substituting the dampened value MBA_{it}' in place of MBA_{it}, I obtained a model of cluster occupancy conditional on prior occupancy:

$$\Pr(X_{it} = 1 \mid X_{i,t\text{-}1} = 1, MBA_{it}) = \text{logit}^{-1}(8.914 - 0.5489\ MBA_{it}). \tag{1}$$

I chose the parameter values in MBA_{it}' subjectively but in such a way that the resulting curve (equation 1) forecasts 0.80 probability of occupancy when midstory basal area is approximately 13.8 m^2/ha (60 ft^2/ac); beyond this level, the curve drops rapidly to 0 (Figure 14). In the original model of Loeb et al. (1992), probability of occupancy is 0.20 at this same level of hardwood midstory density.

Midstory basal area was not a quantity measured in vegetation surveys or tracked in the simulation model, nor is it routinely measured by Refuge foresters in compartment management assessments (J. D. Metteauer, Piedmont National Wildlife Refuge, personal communication). To permit use of the woodpecker persistence model, I transformed computed values of understory vegetation density, V (see *Understory Vegetation Density*, above), into surrogate values for midstory basal area. I lacked information on the nature of the relationship between *MBA* and V. I therefore assumed that $V^{2/3}$ was directly proportional to *MBA* and that a positive correspondence could be formed between the two quantities on the basis of Refuge-wide summary statistics. From results reported by Loeb et al. (1992), who summarized data collected at the Refuge in 1984, I calculated *MBA* mean ($\bar{x}_{MBA}$ = 9.174 m^2/ha) and variance (s^2_{MBA} = 46.55) pooled over active and inactive clusters. I compared these statistics with those that I calculated for $V^{2/3}$ ($\bar{x}_V$ = 85.13 [kg/ha]$^{2/3}$; s^2_V = 6807) from all old ($\geq$80 years) pine stands of moderate density (11.48-16.07 m^2/ha [50-70 ft^2/ac]) in simulation year 2000. The transformation that I used,

$$M\hat{B}A_{i,t} = \hat{V}_{i,t}^{\ 2/3}\left(\frac{\bar{x}_{MBA}}{\bar{x}_V}\right),$$

scaled the new values so that their Refuge-wide mean and coefficient of variation matched those summary statistics obtained from the 1984 midstory basal area measurements.

The simulation model checked every map cell containing a woodpecker cluster. If the pine stands in the cell had been previously destroyed by random disturbance, the procedure changed cluster status of the cell from “present” to “absent.” If not, and if the cluster was active in the previous year, the program computed the estimated value of MBA_{it} for the cell and the probability $\Pr(X_{it} = 1 \mid X_{i,t-1} = 1, MBA_{it})$ (equation 1). The program drew a random value from a uniform distribution and classified cell *i* as “active” in time period *t* if the calculated probability value exceeded the random value or “inactive” otherwise.

Red-cockaded woodpecker populations are limited by availability of nesting habitat (Walters 1991), and establishment of recruitment clusters is the principal means by which the population of breeders may be increased (Walters 1991, Heppell et al.

1994). Quality of nesting habitat, proximity to active clusters, and ample recruitment into the population are probably the most important factors influencing settlement of inactive clusters (Doerr et al. 1989, Thomlinson 1995, Thomlinson 1996, Azevedo et al. 2000).

I extended the Loeb et al. (1992) model to produce a model of settlement sensitive to both habitat quality and reproductive isolation (Figure 15):

$$\Pr(X_{it} = 1 \mid X_{i,t\text{-}1} = 0, MBA_{it}, R_{it}) = \text{logit}^{-1}\{5.134 - 0.4574\, MBA_{it} + a(R_{it} - R_0)\, I(R_{it} > R_0)\}, \qquad (2)$$

where $I(x)$, the indicator function of x, is 1 when x is true and is 0 otherwise. In this model, a and R_0 were additional parameters and R_{it} was a measure of reproductive isolation (km^2/recruit) for cluster i at time t. I modified the isolation coefficient used by Thomlinson (1995) to reflect reproductive isolation from all J active clusters in the population at time t:

$$R_{it} = \frac{1}{\sum_{j=1}^{J} \frac{r_{jt}}{\rho_{ij}^{2}}}\ .$$

Here r_{jt} was recruitment from active cluster j at time t and ρ_{ij} was distance (km) from cluster i to active cluster j.

I derived a probability distribution for recruitment per active cluster, r_{jt}, from field data collected at the Refuge since 1990 (J. A. Mason, Piedmont National Wildlife Refuge, personal communication) and from literature sources. The Refuge's woodpecker productivity surveys yield annual counts of nestlings (n_{jt}), but fledglings (f_{jt}) have not been counted since 1995. I calculated a nestling group size distribution ($p_n(x)$) by pooling 1990-2000 counts of nestlings produced per active cluster (Table 2). I also cross-tabulated nestling group size by fledgling group size using pooled data from 1990-1995 (Table 2). From this table, I calculated empirical probabilities of fledgling group size conditional on nestling group size ($p_{f|n}(x \mid n)$, Table 2). I calculated the unconditional fledgling group size distribution by multiplying the conditional probabilities by their corresponding nestling group size probabilities, then summing over all nestling group size classes (Table 2):

$$p_f(x) = \sum p_{f|n}(x \mid j) p_n(j).$$

Using the estimated $\phi = 0.38$ rate of fledgling survival to adulthood (Maguire et al. 1995) and assuming that fledgling fates were independent, I calculated recruitment group size distribution per active cluster (Table 2):

$$p_r(x) = \phi^x \sum_{i=x}^{4} p_f(i) \binom{i}{x} (1-\phi)^{i-x} .$$

The computer model drew stochastic integer values of r_{jt} for all active clusters at time t based on the probability distribution $p_r(x)$. Then for each inactive cluster i, the program computed the value of R_{it} and provided it into the settlement model (equation 2) along with the cell value of MBA_{it}. In the model, I chose parameter values $a = -0.691$ and $R_0 = 0.0810$. I made both selections subjectively, however, because I lacked data to estimate these values. The value $R_0 = 0.0810$ corresponded to the availability of two recruits in the population, both within a 402-m radius of the cluster site, and implied that the cluster site is effectively not reproductively isolated in this circumstance (i.e., the effect of reproductive isolation on settlement probability was zero for any smaller value of R). The coefficient a was negative, reflecting the negative relationship between reproductive isolation and settlement probability for a fixed value of habitat quality. The value $a = -0.691$ corresponded to 0.01 probability of settlement for an inactive cluster with low midstory density ($MBA = 5.74$ m^2/ha [25 ft^2/ac]) but with moderately high reproductive isolation ($R_{it} = 10.37$, or equivalent to one recruit in the population at a distance of 3220 m). I compared the probability value returned by the model to a random value drawn from a uniform distribution. If the probability value exceeded the random value, the status of the inactive cluster was changed to "active"; otherwise, the cluster was left as "inactive."

The HEF provides foraging habitat for woodpeckers on the Refuge, serves as a dispersal corridor between the northern and southern portions of the Refuge, and produces birds that may ultimately settle in the Refuge. Managers of the ONF and HEF conduct cluster monitoring annually, but because birds are not banded, population recruitment and dispersal cannot be estimated. The 1999 survey of the ONF yielded 18 active clusters, most occurring in the HEF (E. D. Caldwell, Oconee National Forest, personal communication). Because the simulation model did not address the mostly unknown habitat and bird dynamics on the HEF, I had no explicit way to incorporate dispersal from HEF to the Refuge. I instead assumed a constant productivity rate of six recruits annually on the HEF, and I assumed all were available at a single cluster located near the center of the HEF. I obtained this recruitment value by assuming a constant population of 12 active clusters in the HEF and by assuming that average rates of fledglings produced per cluster (1.32) and fledgling survival to adulthood (0.38) computed for the Refuge also apply to the HEF.

One of the significant uncertainties facing managers of red-cockaded woodpecker populations is the relationship between quality or quantity of foraging habitat and woodpecker productivity and recruitment. The first recovery plan for the species emphasized adherence to minimum constraints on amounts and distributions of stem sizes and ages within 805-m (½-mile) foraging buffers of clusters (U.S. Fish and Wildlife Service 1985). Ligon et al. (1986) criticized the threshold values chosen, pointing out that they were based on study of a single population in favorable habitat. Beyer et al. (1996), Wigley et al. (1999), and studies cited therein found no association between amount of foraging habitat and productivity. On the other hand, Conner et al. (1999) and Davenport et al. (2000) reported relationships between foraging habitat quality and group fitness.

I proposed an alternative parameterization (model W1) to the base recruitment model (model W0) that allowed amount of foraging-quality habitat (pine ≥40 years old, 9.18-18.37 m^2/ha basal area) to influence the probability distribution of fledgling group size, $p_f(x)$, and rate of fledgling survival to adulthood, ϕ. I calculated $\psi(h_i)$, a scaled habitat quantity measure as

$$\psi\,(h_i) = \zeta\left(\frac{h_i - h'}{h'}\right) + 1\,,$$

where ζ was a scaling coefficient, h_i was the amount of foraging-quality habitat within a circular 49-cell neighborhood (198.5 ha) of active cluster i, and h' was a central habitat quantity. Where the circular neighborhoods included off-Refuge lands, habitat contributions by ownership type were approximated as before (see *Thinning*, above). The transformation provided a habitat effect multiplier for $p_f(x)$ and for ϕ. When $\zeta > 0$, habitat quantity exceeding the central value h' yielded $\psi(h_i) > 1.0$, and habitat quantity falling below h' provided $\psi(h_i) < 1.0$. I used a central value of 75 ha for h'.

I applied the habitat quantity multiplier to the fledgling group size distribution to produce a habitat-influenced distribution of fledgling group sizes:

$$p_f'(x_i) = \frac{p_f(x_i)\psi(h)^{(x_i - f')/f'}}{\sum_j p_f(x_j)\psi(h)^{(x_j - f')/f'}}.$$

In this model, the probabilities $p_f(x_i)$ were enhanced or reduced according to the value of the habitat multiplier. The exponent $(x_i - f')/f'$ modified the multiplier such that probability enhancements (or reductions) for large values of x_i were offset by reductions (enhancements) in probability for small values of x_i. I chose the 1990-2000 estimated mean rate of fledgling production at the Refuge, 1.32, as the central value f'. The denominator of this model simply served to scale the habitat-modified probabilities $p_f'(x_i)$ so that they added to 1.0. I used the value $\zeta = 0.8$ in the alternative model W1. This value permitted average fledgling production per active cluster to vary from 0.20 to 2.29 for the poor and excellent habitat extremes, respectively (Figure 16A).

I also used the habitat quantity multiplier to modify fledgling survival to adulthood:

$$\phi' = \text{logit}^{-1}\{\log(\phi/(1-\phi)) + \alpha(\psi-1)\}.$$

Here, α was a parameter controlling the strength of the habitat influence, and I fixed this value at $\alpha = 2$ in the alternative model. Though my choices of ζ and α were subjective, they allowed ϕ' to vary over a reasonable range (0.11-0.90) among poor and excellent habitats (Figure 16B).

I computed a habitat-influenced distribution of recruitment group size per active cluster by substituting the values $p'(f)$ and ϕ' in the model for $p(r)$:

$$p_r'(x) = (\phi')^x \sum_{i=x}^{4} p_f'(i)\binom{i}{x}(1-\phi')^{i-x} .$$

Note that $p_r(x) = p_r'(x)$ when $\zeta = \alpha = 0$. Thus, the original model W0 is a specific case of the general habitat-influenced model W1.

When I simulated model W1, the procedure, as before, drew random integer values of recruitment for the active clusters, but it drew from the alternative recruitment group size distribution $p_r'(x)$. From this point, computation of cluster settlement proceeded exactly as before.

Wood Thrush

Unlike the population model for red-cockaded woodpecker, and unlike the approach taken by Powell (1998), I modeled wood thrush population density in a fully habitat-correlative, non-spatially explicit manner. Wood thrushes are not nearly as habitat-limited as are red-cockaded woodpeckers; by comparison their habitat affinities are quite plastic within mature forest conditions (Hamel et al. 1982, Roth et al. 1996). A wood thrush population would not likely be as sensitive to a distribution of targeted, small-scale management actions as would a population of woodpeckers. Furthermore, data on wood thrush were only available at the compartment-summary level and only for general cover types (pine and hardwood). This was in contrast to the type of data collected for woodpeckers and their associated habitats, available at the resolution of the individual cluster. Therefore, it seemed reasonable to model wood thrush density without spatially-explicit reference, and without individual or territory-level process detail.

My challenge in modeling this system was to reconcile scale differences between compartment-level bird density estimates, habitat data measured at intervals on fixed survey routes, and predicted habitat features available through the landscape simulation model. I developed a two-stage set of statistical models. The first stage related compartment-wide estimated wood thrush densities to compartment-averaged field measurements. The second stage associated the field measurements with landscape cell features averaged at the compartment level. Thus, the linkage between the landscape model and predicted wood thrush densities was indirect in this approach.

Although wood thrush density was often separately estimable by forest overstory type within a compartment, the landscape model simulated habitat characteristics (e.g., basal area, canopy closure, understory vegetation density) only within the pine type. Therefore, I developed two first-stage models. For each model, I used within-compartment averages of pine habitat attributes as predictors for wood thrush density. However, in one model, the response variable was density of wood thrush in the compartment's pine habitats. In the other, the response was density in hardwood habitats.

I restricted my analysis to bird and vegetation data collected in 1998 and 1999, as surveys conducted in those years were most consistent and provided the greatest number of valid compartment-level wood thrush density estimates. Based on patterns I observed in simple scatter plots, I selected four vegetation variables (see *Vegetation Surveys*, Chapter 3) for analysis, all obtained from pine stands 16 years of age or older: BA, the basal area of softwoods, CC, the degree (0.0-1.0) of canopy closure, DB12A, the average of the proportions of cells covered by vegetation in the 0.0-0.3 m and the 0.3-1.0 m ranges of the density board, and DB12L, the logit (log-ratio) of these two proportions. The latter two variables reflected overall vegetation coverage and vegetation structure near ($\leq$1.0 m) the forest floor. I obtained average values of all vegetation variables at the compartment level.

I used simple linear regression in conducting all analyses. Because the distribution of wood thrush density was highly right-skewed, I instead analyzed log(density) as the regression response variable. Furthermore, because each density value was statistically estimated and therefore conveyed a unique level of precision to the analysis, I weighted each response by the coefficient of variation of density, a value approximately equal to the inverse of the standard error of log(density).

I first fit a model containing all four of the vegetation effects to each set (pine and hardwood habitats) of density responses to screen for influential or outlying data points. For the density response in hardwood habitats, I found no evidence of any outlying or influential observations. However, I found a single observation of wood thrush density in pine habitat (1998, compartment 12, $\hat{D}$ = 0.0102) that was much smaller in residual value (Studentized residual = -3.15) than any other value. In subsequent analyses of the density response in pine habitat, I excluded this observation.

To each set of responses, I fit all possible linear models formed from the four main effects and the six two-way interactions. Of models containing interaction terms, I considered only those that also included constituent main-effect terms of the interactions. I calculated the AIC score (Akaike 1973, Burnham and Anderson 1998) for each model, and I selected the superior model on the basis of the lowest AIC score.

The best model for wood thrush density in hardwood habitats, based on attributes measured in neighboring pine stands, was

$$\log(\hat{D}_H) = 18.89 - 0.4544\ BA - 26.77\ CC - 1.823\ DB12A + 0.5539\ BA{\times}CC$$

($R^2 = 0.531$, AIC = 5.524, dfe = 11). Explanatory variables in this model were highly collinear. Thus, it was not surprising that none of the effects were large relative to their standard errors (SE = 22.39, 0.3341, 24.91, 1.314, 0.3747, respectively). For given compartment values of pine habitat BA, CC, and DB12A, I predicted values of wood thrush density in hardwood habitat, corrected for back-transformation bias, as follows:

$$\hat{D}_H(BA, CC, DB12A) = \exp(\ \log(\hat{D}_H) + \sigma_{\hat{D}}(H)^2/2\ ),$$

where $\sigma_{\hat{D}}(H)^2$ was the estimated variance of the predicted conditional mean for the response.

The model implied that wood thrush density in hardwood habitats increased as total understory vegetation coverage in neighboring pine habitats decreased, for example, through burning. The interaction between basal area and canopy closure suggested that bird density in hardwood habitats was not responsive to pine overstory conditions unless those conditions were dense: when pine basal area and canopy closure were very high ($BA \geq 18.4\ m^2/ha$, $CC \geq 0.90$), wood thrush density sharply increased in hardwood habitats.

For the response in pine habitats, the best linear model was

$$\log(\hat{D}_P) = -7.803 + 0.09640\ BA - 4.658\ DB12A - 14.81\ DB12L + 0.2518\ BA{\times}DB12L - 5.997\ DB12A{\times}DB12L$$

($R^2 = 0.797$, AIC = -15.02, dfe = 9). In contrast to the model for the response in hardwood, parameter estimates were large relative to their standard errors (SE = 1.631, 0.02757, 0.9824, 4.705, 0.07882, 2.880, respectively). Wood thrush densities in pine habitats responded positively to pine basal area, and the magnitude of this relationship became greater as floor-level vegetation increased relative to mid-level vegetation (larger values of DB12L). Densities in pine habitats decreased with total vegetation coverage (DB12A), and the negative relationship became more pronounced with greater values of DB12L. To back-transform estimated densities of wood thrush in pine habitats given values of BA, DB12A, and DB12L, I used the following formula:

$$\hat{D}_{\mathrm{P}}(\mathrm{BA},\ \mathrm{DB12A},\ \mathrm{DB12L}) = \exp(\ \log(\hat{D}_{\mathrm{P}}) + \sigma_{\hat{D}}(\mathrm{P})^2/2\ ),$$

where $\sigma_{\hat{D}}(\mathrm{P})^2$ was the estimated variance of the predicted conditional mean for the response.

Thus, models for wood thrush density in each habitat type were conditional on compartment averages of pine basal area, canopy closure, and two understory vegetation measures. The landscape simulation model tracked stand-level values of basal area and canopy closure. Therefore, I obtained values of BA and CC for the wood thrush models directly by computing area-weighted averages of these quantities by compartment for all stands ≥16 years old.

I built the second-stage models to provide estimated values for DB12A and DB12L given compartment-level summaries of forest variables from the landscape model. I synchronized the simulation model to each of the years 1998 and 1999 using the approach described earlier (see *Forest State Synchronization*, Chapter 3). I computed stand-level values of basal area, canopy closure, time since last burning (TSLB), and understory vegetation density (V), and I averaged these values by compartment for each year.

I used ordinary least squares regression to find associations between simulation-derived habitat conditions and the field-measured values of DB12A and DB12L. I fit all possible models formed by combinations of BA, CC, TSLB, and V to each of the responses. I considered only main-effects models and did not consider interactions between variables. I computed AIC for each model and selected models yielding lowest AIC.

The top-ranked model for DB12A (AIC = -55.7) contained only the variable TSLB ($P = 0.020$):

$$\hat{y}_{\mathrm{DB12A}} = 0.3482 + 0.02332\ \mathrm{TSLB}\ (R^2 = 0.43,\ \mathrm{dfe} = 10).$$

Thus, the average of the vegetation coverage proportions measured by the lowest two sections of the density board was positively related to number of years since burning. The next-highest ranked model (AIC = -54.5) contained positive and negative terms in TSLB and CC, respectively.

The best model for DB12L (AIC = -42.7) contained the variables TSLB ($P = 0.146$) and V ($P = 0.474$):

$$\hat{y}_{\mathrm{DB12L}} = -0.5966 + 0.03339\ \mathrm{TSLB} + 1.554\times10^{-4}\ \mathrm{V}\ (R^2 = 0.46,\ \mathrm{dfe} = 9).$$

In this model, the coverage of floor-level vegetation (0.0-0.3 m) relative to mid-level vegetation (0.3-1.0 m) increased with time since burning and with estimated overall vegetation density. TSLB and V were moderately correlated ($r = 0.60$). Thus, the overall model fit is stronger than is suggested by the precision estimates for the variables.

Thus, I used compartment-averaged values of V and TSLB from the simulation model to predict values of DB12A and DB12L. These predicted values, in conjunction with BA and CC calculated from the simulation model, then became input to the wood thrush density models. I calculated compartment-level abundance of wood thrush from the estimated quantities $\hat{D}_P$ and $\hat{D}_H$ and from total area in pine and hardwood habitats.

The models were developed on the assumption of linearity between responses and explanatory variables, but this assumption must ultimately fail at some unknown point beyond (or possibly within) the range of the calibration data. Therefore, I proposed an alternative structural form to this base model (model T0). For each of the $\hat{D}_P$ and $\hat{D}_H$ model inputs, I specified lower and upper response limits corresponding to the range of field data used to fit the models. In the alternative model (model T1), an input value lower than its lower field-measured limit or greater than its upper field-measured limit was replaced by the limit value. The modified value was supplied to the model set as if it was the measured value. This action effectively forces abrupt nonlinearities into the model at the data extremes. Above (or below) an input value extreme, wood thrush density no longer responds to further increases (or decreases) in the variable.

PARAMETER SELECTION

Behavior of the Refuge management model was governed by 118 parameters (Table 1). I chose values for many parameters subjectively, but all had some degree of empirical, statutory, or conceptual support. Rationale for most of my choices appears earlier in this chapter.

Despite whatever sound rationale may exist for the parameter values used in this model, considerable uncertainty about parameter values exists nonetheless. Sensitivity analysis, which portrays sensitivity of model output to perturbations in the parameters, is out of the question because of the many parameters, the model’s long execution time, and output stochasticity. However, field data measurements of birds may be used as a crude validation of the model; validation and model updating are treated below.

MODEL SIMULATION UNDER DECISION ALTERNATIVES

As described above, the stand-level silvicultural actions carried out in the model are rule-based and are conditional on the current status of the forest and the red-cockaded woodpecker population distribution. That is, current conditions dictate to the manager, through fixed rules, a set of tactical operations concerning exactly where, in what manner, and how much to cut.

I believed that enforcing fixed, rule-based actions in the simulation model was necessary for two reasons. First, although in practice managers routinely alter stand cutting actions on a case-by-case basis, managers are nevertheless cognizant of and tend to adhere to fundamental action rules (e.g., minimum distance from active woodpecker cluster required for placement of recruitment cluster). Second, management alternatives available to the Refuge are somewhat constrained as statutory and recovery plan obligations often dictate the range of alternatives and many of the decisions. I saw no reason to further complicate the model by incorporating softer or dynamically varying rules.

Rather than place focus on these more tactical types of actions, I confined the search for optimal management decisions within the more strategic context of current Refuge management. Specifically, for a given population objective, I hoped to find the following using the simulation model: (1) an optimal compartment permutation sequence, and (2) an optimal compartment burn limit.

The compartment permutation sequence specifies the assignment of compartments into management groups and is one of the inputs to the model. Currently, the Refuge uses the following permutation sequence (Figure 6):

Group	Compartments
1	1, 8, 13, 24, 32
2	9, 12, 21, 25
3	10, 27, 30, 33
4	6, 14, 23, 29
5	4, 7, 18, 22
6	3, 5, 15, 20
7	11, 17, 26, 34
8	2, 16, 19, 28, 31

The permutation sequence dictates the order of compartments that the simulation model visits through time for silvicultural operations. All compartments in a single group are processed in one simulation year. After processing group 8, the simulation program returns to group 1 compartments in the next time period. Any of the groups may be specified for the initial time period. The simulation model cycles through the compartment groups until reaching the chosen time horizon. Given the spatial nature of red-cockaded woodpecker population dynamics, the woodpecker population response may depend on compartment visit order.

The compartment burn limit is a user-supplied value that specifies the number of compartments to burn annually. In each simulation year, the burning submodel ranks the compartments by understory vegetation conditions and burns the highest-ranked compartments up to the compartment burn limit (see *Burning*, above). Inverting the burn limit value and multiplying by 34 provides an average measure of compartment burn periodicity. Burn periodicity may be influential for populations of both red-cockaded woodpeckers and wood thrushes.

Additional parameters provide control over the stochastic features of the model. One parameter sets the number of iterations of the model, so that returns for decision alternatives may be examined in terms of their expected values. The other parameter sets a seed value for the random number generator, allowing the user to control the random number stream.

SELECTION OF DECISION ALTERNATIVES

There are $>10^{26}$ ways of grouping 34 compartments into eight management groups, assuming that two of the groups contain five compartments and the remainder contain four as shown above. However, the number of valid permutations in which no compartment in a single management group is physically adjacent to another (by "adjacency" I mean sharing a common linear boundary) is unknown. I wrote a computer program to randomly search compartment permutations and to extract 10,000 permutations that satisfied the criterion of non-adjacency within management groups (Appendix C.13). I found, on average, a valid permutation in every 1328 trials, leading me to estimate that there exist 8.1×10^{22} permutation sequences that meet the non-adjacency criterion.

I first programmed a genetic algorithm to search for near-optimal permutation sequences to maximize red-cockaded woodpecker and wood thrush population objectives. Despite the fact that the number of qualifying permutations was only a fraction of the number of all permutations, even this number of decision alternatives is overwhelming for any search algorithm, particularly when evaluation of each decision could take several minutes to possibly a few hours. Unfortunately, this was also the case for the genetic algorithm approach, and I was forced to abandon it. I instead elected to use simulation to evaluate a much smaller, manageable number of permutation alternatives that I chose judiciously from the sample of 10,000 valid alternatives.

I used the GIS to locate the geographic center of each compartment. Given a set of compartments in a management group, I calculated average inter-compartment distance for the group. For a single permutation sequence, I averaged these mean distances over all the groups in the sequence to obtain a measure of within-year spatial dispersion. I also computed average distance between compartments in temporally adjacent groups (i.e., compartments where visits were separated by one year in time) for a measure of successive-year spatial dispersion (Appendix C.14).

I ranked the entire set of 10,000 permutation sequences on the basis of within-year spatial dispersion and successive-year spatial dispersion. For each permutation, I computed the average of the two ranks and the difference between the ranks. I chose for analysis the sequences with the two highest average ranks and the sequences with the two lowest average ranks. In the first case, compartment selections were maximally dispersed both within years and between successive years. In the second case, just the opposite was true: compartment selections were minimally dispersed both within year and between successive years. I also chose the sequences with the two highest and the two lowest rank differences. In one case, compartment selections were maximally dispersed within years but minimally dispersed between successive years, and the converse was true for the other case. Thus, the eight permutation sequences that I chose provided extremes in spatial dispersion of compartments both within time periods and between time periods. Although simulation of this set of permutation sequences does not constitute a true optimization study with respect to all possible sequences, I believed that if an optimum does exist for any objective function, it is likely to occur for a sequence that resembles one of the eight extremes. To this set of eight, I added the Refuge's operational permutation sequence (Figure 6) for comparison.

I simulated each of the nine permutation sequences in combination with two compartment burn limit settings. I chose burn limits of seventeen compartments, representing an average two-year burn cycle, and seven compartments, representing an average five-year burn cycle. The latter setting is approximately consistent with the level of burning currently employed at the Refuge (4-9 compartments [$\geq$40.5 ha] burned annually 1991-2000, $\bar{x} = 6.1$ compartments).

SIMULATION AND ANALYSIS OF DECISION ALTERNATIVES UNDER MODEL UNCERTAINTY

I simulated the decision alternatives under different model scenarios and conducted analyses to investigate patterns in optimal decisions. My analysis goals were to (1) find optimal management actions for bird population objectives under each alternative model of forest and bird dynamics, (2) determine whether optimal actions differed among the uncertain models and among objective functions, and (3) estimate the expected benefit in reducing model uncertainty.

I formed twelve alternative models from combinations of the three submodels of forest dynamics (F0, F1, F2), the two submodels of woodpecker response (W0, W1), and the two submodels of wood thrush response (T0, T1). I denoted each model as M_{FWT} where the subscripts take on values denoting the specific submodel contained.

Thus, I simulated eighteen decision alternatives under each of the twelve alternative models. I ran each model-decision combination over a 100-year time horizon beginning from estimated forest conditions and measured woodpecker conditions at year

2000. I recorded 100-year compartment-level totals of active red-cockaded woodpecker clusters and wood thrush abundance. Because the simulation models were stochastic, I repeated each run twenty times.

For each model alternative, I used analysis of variance to test effects of the decision alternatives on final total population size of active clusters and final total population size of wood thrushes. Additionally, I analyzed a composite measure of the two abundance quantities. I scaled each abundance quantity by its standard deviation computed from all (360) replicate runs for each model, then I calculated the simple average of the scaled values.

The experimental design was in the form of a split-plot. Permutation sequences served as the main-plot experimental units. I analyzed levels of within-year spatial dispersion (DWT) and successive-year spatial dispersion (DBT) in a factorial arrangement at the main-plot level. Because I assigned burning periodicity after selecting the set of permutation sequences, levels of compartment burn limit (BURN) served as the split-plot effect. I obtained estimates of treatment level means and standard errors following this analysis. To preserve analytical balance, I excluded the Refuge operational permutation sequence from the split-plot analysis. However, I obtained mean outcomes for the operational sequence and compared them to the means for the other treatment levels.

Lastly, I ranked decision alternatives according to model-averaged measures of the three population objectives. For each model and decision alternative, I multiplied the mean outcome by a model confidence weight, then I added the products to yield a model-averaged score for the decision. To reflect complete uncertainty among models, I assigned a confidence weight of 1/12 to each model. The top-ranked decision alternative for each objective was my assessment of the optimal management action under uncertainty about bird and forest response to management. From these data, I computed value of information, expressed in units of the resource, for each objective response (Lindley 1985:120, Dakins 1999).

MODEL VALIDATION AND MODEL UPDATING

Each alternative model provided a predicted distribution of active woodpecker clusters (cell level) and wood thrushes (compartment level) in response to simulated management actions. In particular, each model provided a prediction of year 2001 bird population status conditional on management actions and forest state in year 2000. The Refuge provided me data on year 2001 woodpecker cluster status and distribution (J. A. Mason, Piedmont National Wildlife Refuge, personal communication). The most recent data on wood thrush abundance, however, were only available for year 2000 as part of the Refuge-wide point count study (W. T. Plummer, University of Georgia, unpublished data).

Because the models were developed independently of the 2001 woodpecker data, this data set provided some basis for a crude validation of each model and for updating confidence weight in each. I simulated each model 1000 times over the single time step from 2000 to 2001. For this analysis, I simulated harvesting and burning activities that occurred on the Refuge between breeding seasons 2000 and 2001. I aggregated model output and field data counts of active woodpecker clusters to the compartment level. I assumed that each predicted average outcome for each compartment specified a mean for a Poisson probability distribution for number of active clusters. Given the mean μ_{ij} in compartment j under model i, I calculated a probability value of active cluster abundance X_j:

$$\Pr(X_j \mid \mu_{ij}) = \mu_{ij}^{X_j}\, e^{\mu_{ij}} / X_j! \,.$$

I assumed the compartment-level cluster abundance outcomes were independent. Thus, I multiplied the $\Pr(X_j \mid \mu_{ij})$ over all compartments j to obtain a total likelihood $\mathscr{L}_i(X)$ for each model i. Note that $\Pr(X_j \mid \mu_{ij})$ is undefined when $\mu_{ij} = 0$, therefore, compartments for which $\mu_{ij} = 0$ were excluded from the calculation.

Given the model-specific likelihoods for response X and given the set of equal prior confidence weights ($p_{0i} = p_0 = 1/12$), I used Bayes' formula to compute posterior confidence weights p_{1i} among models:

$$p_{1i} = \frac{p_{0i}\mathscr{L}_i(X)}{\sum_{k=1}^{12} p_{0k}\mathscr{L}_k(X)} \quad .$$

Thus, belief weight was redistributed among models in response to the confrontation of each model to monitoring data.

CHAPTER 5

RESULTS

WOOD THRUSH SURVEYS

Wood thrushes were commonly encountered throughout the Refuge in all years and under both hardwood and pine overstories. Densities of wood thrush were generally greater in 1999 than in 1998, but no overstory habitat type contained consistently greater densities than the other (Table 3). Range of density estimates (0.0067-0.1084 birds/ha) was consistent with, but lower than, the range (0.0150-0.1302 males/ha) reported by Powell et al. (2000), and both were well below the maximum density (1.0 territories/ha) encountered in certain other habitats (Roth et al. 1996). Populations of wood thrush were somewhat less dense in habitats that had recently been burned than in those that had not (Final Report; W. T. Plummer, University of Georgia, unpublished data).

OVERSTORY TRANSITION MODEL

OPTIMIZATION

Program ASDP required 179 hours to evaluate 500 stage iterations of the overstory transition model. Despite the effort, the program did not converge on a stationary optimal decision policy for maximizing amount of old-growth forest habitat. However, by the end of the iterations, decision values between successive iterations were changing in fewer than ten of the 14,641 (0.07%) state value combinations. Thus, the policy that resulted (Appendix D) was likely very similar in appearance to a stationary optimal policy.

SENSITIVITY ANALYSIS

Each response of the overstory transition model was highly sensitive to perturbations in at least one of the parameter values (Table 4). Varying any parameter that controlled the rate of transition between two pine age classes (τ_{21}, τ_{32}, τ_{43}, τ_{14}) produced a large, negative estimate of elasticity for the donor age class and large, positive estimates of elasticity in the receiving age class and all older classes. I found these patterns at all levels of the adjusted parameter. The upland hardwood response variable was mostly unaffected by perturbations in the age class transition parameters.

Elasticities for the pine class responses were negative with respect to the hardwood encroachment parameters ($\mu(e_i)$) and the encroachment rate variability parameter (v_e), but elasticities for the upland hardwood response variable for these same variables were positive (Table 4). Magnitude of elasticity was larger when the perturbed parameter was tested at high rather than low levels (Table 4).

Of all responses, elasticities for the P4 age class were the greatest with respect to parameter k, which controlled the amount of forest successfully entering the P4 state (Table 4). For the parameter controlling the amount of hardwood reverting to pine (q), I found large negative values of elasticity for the upland hardwood response (Table 4).

With respect to the parameter controlling mean disturbance rate, $\mu(c)$, positive responses in elasticity by the P1 and P2 age classes were matched by negative responses for the upland hardwood class (Table 4). Largest magnitudes of the elasticity response occurred for the largest value of $\mu(c)$.

All of the model responses were highly sensitive to perturbations in the variance of the disturbance rate ($\sigma^2(c)$), and magnitude of the elasticity responses increased with increases in the parameter value (Table 4). Elasticity responses by the pine classes were positive, and those by the hardwood class were negative.

The optimal harvest decision policy for the woodpecker nesting habitat objective was highly sensitive to alternative model scenarios (Figure 17, Appendix D). For model F1, ASDP failed to converge on a stationary policy after 500 stage iterations, though the policy derived was likely similar to a stationary policy (<0.1% of all state value combinations changed in decision value between successive iterations). However, ASDP did arrive at a stationary policy for model F2 after 214 iterations. Under optimal policies for either model F1 or model F2, harvesting in all age classes was less aggressive overall than for the policy corresponding to the base model F0 (Figure 17).

I used the synchronized GIS to estimate year 2000 forest type distribution. For this distribution and for the objective of maximizing woodpecker nesting habitat, I determined optimal regeneration decisions under each of the three models. Choice of pine types targeted for cutting was highly dependent on choice of the appropriate forest dynamics model. In particular, model F1 (low rate of transition to P4) indicated extensive harvesting of the P3 type whereas model F2 (high rate of transition to P4) suggested concentration of harvest in the P2 type:

	Year 2000 Estimated Cover		Optimal Harvest Decisions					
			Base Model F0		Model F1		Model F2	
Cover Type	Area (ha)	%	% of type	Area (ha)	% of type	Area (ha)	% of type	Area (ha)
P1	385	3						
P2	1924	15	4	77	0	0	14	269
P3	7090	57	2	142	12	851	0	0
P4	828	7	4	33	6	50	0	0
UH	2319	18						
Total	12546	100		252		901		269

I was unable to obtain comparable regeneration statistics from compartment prescription data for the Refuge. However, during the decade 1989-1998, the average area of pine habitat regenerated into the P1 type was 53.6 ha (J. D. Metteauer, Piedmont National Wildlife Refuge, personal communication), a substantially smaller total regeneration amount than any estimated as optimal by these models.

Spatially-explicit Forest Management Model

Selection of Permutation Sequences

I found eight permutation sequences that provided high contrast in between-year and within-year inter-compartment distance measures (Table 5). In comparison, distance measures for the Refuge's current operational sequence were intermediate. Ranks for the operational sequence were extreme for neither the within-year (5051, 50.5%) nor the successive-year (3602, 36.0%) dispersion measures (Table 5).

Analysis of Decision Optima

I ran 4320 simulations (12 model alternatives × 9 permutation sequences × 2 burn scenarios × 20 replications) using 519 processor-hours of computer time. Mean 100-year populations sizes were 123 active red-cockaded woodpecker clusters (range 10-246) and

4.28×10^{13} wood thrushes (range 253-6.17×10^{16}). Predicted wood thrush abundances were right-skewed, and, for one class of prediction models, were implausibly large. I therefore doubly log-transformed (log-log-transform) wood thrush abundances in all subsequent analyses. The stochastic nature of the model is clearly evident in a sample of 20 runs for the Refuge's operational permutation sequence (Figure 18).

Red-cockaded Woodpecker Response

Numbers of active red-cockaded woodpecker clusters were highly sensitive to the decision alternatives, and the patterns of variation differed among simulation models (Table 6). Under every model, cluster number responded strongly to the maximum compartment limit imposed on burning (effect BURN, $P \leq 0.001$, Table 6). On average, burning seventeen rather than seven compartments annually produced 80 more active clusters at 100 years (Table 7), and the direction of the difference was consistent among models.

I found evidence in many models that cluster numbers also responded to compartment permutations. Choice of forest overstory submodel more than any other type of submodel appeared to have greatest influence in the pattern of variation with respect to permutations. Among the four models of moderate transition to the P4 type (models $M_{0\cdot\cdot}$), average number of clusters was greater when successive-year compartment dispersion was small rather than large (Table 7), though evidence for these differences was slight (effect DBT, $0.104 \leq P \leq 0.226$, Table 6). This effect was not important in any other model ($P \geq 0.413$, Table 6) except in model M_{200} ($P = 0.024$, Table 6), which predicted a high rate of transition into the P4 type. Under this model, cluster number was greater when successive-year compartment dispersion was high rather than low (Table 7). Though differences were not significant, greater mean number of clusters tended to occur when successive-year compartment dispersion was large for the models in class $M_{1\bullet\bullet}$ (low rate of transition to P4) and when dispersion was low for models in class $M_{2\bullet\bullet}$ (Table 7).

Within-year compartment dispersion (effect DWT) influenced cluster outcome, and its influence also varied mostly among forest overstory submodels. The effect was strongest among the models in class $M_{2\bullet\bullet}$ ($0.016 \le P \le 0.229$, Table 6). In all cases for this model class, mean number of clusters was greatest when within-year compartment dispersion was high rather than low (Table 7). In the other overstory model classes, mean cluster numbers tended to be greatest under low within-year dispersion for models in class $M_{0\bullet\bullet}$ and under high within-year dispersion for models in class $M_{1\bullet\bullet}$ (Table 7).

The interaction between the two types of compartment dispersion (effect DBT × DWT) was weakly to strongly influential on mean cluster numbers under models M_{000}, M_{001}, M_{011}, M_{200}, and M_{201} ($0.003 \le P \le 0.188$, Table 6). The treatment level corresponding to the greatest mean response was consistent in ten of the twelve models: low degree of successive-year compartment dispersion and high degree of within-year dispersion (Table 7). However, the treatment level corresponding to the lowest mean response differed among models but was generally consistent within overstory submodels. For all models in the class $M_{0\bullet\bullet}$, the lowest mean response occurred for the combination of high successive-year dispersion and high within-year dispersion (Table 7). For the other two classes, the lowest mean response occurred most often for the combination of low successive-year dispersion and low within-year dispersion (Table 7).

The overall woodpecker response mean was highly sensitive to the choice of overstory model. The overall mean was greatest for the $M_{2\bullet\bullet}$ class ($\bar{x} = 138.4$, SE = 0.95), lowest for the $M_{1\bullet\bullet}$ class ($\bar{x} = 107.4$, SE = 0.42), and intermediate for the $M_{0\bullet\bullet}$ class ($\bar{x} = 124.0$, SE = 0.47) (Table 7). With regard to the woodpecker submodels, however, the difference in mean responses for the set of habitat-insensitive productivity submodels $M_{\bullet 0 \bullet}$ ($\bar{x} = 124.0$, SE = 0.54) and the set of habitat-sensitive submodels $M_{\bullet 1 \bullet}$ ($\bar{x} = 122.6$, SE = 0.54) was statistically significant ($P = 0.067$) but biologically trivial.

Though the overall means varied by model, optimal decision alternatives for the woodpecker response were mostly unaffected by model choice (Table 8). In ten of the twelve models, the optimal decision alternative was that of frequent burning, combined with a compartment visit schedule that maximized within-year compartment dispersion and minimized successive-year dispersion ($\bar{x} = 168.3$, Table 8). For all models, infrequent burning was always suboptimal, but the worst decisions with regard to compartment permutations depended on choice of model. For the model class $M_{0\bullet\bullet}$, high between-year and high within-year compartment dispersions were almost uniformly the worst decision ($\bar{x} = 79.8$, Table 8), whereas low levels of both types of dispersion were usually worst for models in classes $M_{1\bullet\bullet}$ and $M_{2\bullet\bullet}$ ($\bar{x} = 78.7$, Table 8).

Relative performance of the Refuge's current operational compartment visitation sequence differed among models and levels of burning (Table 8). In the case of infrequent burning, the operational permutation was inferior to all other permutations in nine of the twelve models (Table 8). When burning was more frequent, the operational sequence was often favorably ranked, but never top-ranked (Table 8).

Under complete uncertainty about choice of the correct model, decision actions utilizing frequent burning were superior to those that did not (ranks for model-averaged outcomes, Table 8). The optimal permutation sequence for the woodpecker response differed with respect to burning periodicity. Under frequent burning, the low-high combination of successive/within-year compartment dispersions was optimal, whereas the high-low combination was optimal under infrequent burning (Table 8). Furthermore, the Refuge's operational permutation was the second-ranked alternative overall, although its expected return was practically indistinguishable from those of the other suboptimal alternatives under frequent burning (Table 8). Under infrequent burning, however, the operational sequence was the worst alternative (Table 8).

Wood Thrush Response

In all but one model (M_{200}), wood thrush abundances were highly sensitive to management decisions, particularly with respect to the frequency of burning ($P \leq 0.086$, Table 6). On average, densities of wood thrush were approximately five times as great in lightly-burned landscapes as in intensively-burned landscapes ($\bar{x}$ difference in log-log abundance = 0.182, Table 7).

Few of the compartment dispersion effects were particularly strong (Table 6). Effect DWT was significant in only one instance (Model M_{211}, $P = 0.050$, Table 6). However, comparison of means for this effect in all models suggested a consistently greater abundance of wood thrushes when within-year compartment dispersion is high rather than low (Table 7). Analysis of means for the interaction effect similarly suggested greater abundances when the DWT component of the interaction is at its high value (Table 7). I saw no pattern in wood thrush abundances with regard to successive-year compartment dispersion (Table 7).

Overall mean of the wood thrush response was strongly dependent on both choice of overstory submodel and wood thrush habitat-association submodel. Mean wood thrush response was greatest for models of class $M_{1\bullet\bullet}$ ($\bar{x} = 2.332$, SE = 0.00618), least for models of class $M_{2\bullet\bullet}$ ($\bar{x} = 1.982$, SE = 0.00185), and intermediate for models of class $M_{0\bullet\bullet}$ ($\bar{x} = 2.120$, SE = 0.00323) (Table 7). Thus, predicted population abundance of wood thrush was positively associated with predicted rate of overstory transition into the hardwood class.

Mean response for the linear habitat-association model class ($M_{\bullet\bullet 0}$, $\bar{x} = 2.295$, SE = 0.00476) was greater than that for the nonlinear model class ($M_{\bullet\bullet 1}$, $\bar{x} = 1.995$, SE = 0.00068) (Table 7). Thus, population abundances corresponding to the linear version of the habitat model exceeded those corresponding to the nonlinear version by more than an order of magnitude.

The two models in the class $M_{1\bullet 0}$, which projected a high rate of overstory transition into the hardwood type and a linear response by wood thrushes to habitat conditions, produced wildly implausible estimates of 100-year bird abundance (maximum projected 100-year total abundance = 6.17×10^{16}). The models predicted average 100-year outcomes of approximately 18,000 (approx. 1.3 birds/ha) and 3×10^{8} (approx. 21,000 birds/ha) birds in frequently-burned and infrequently-burned landscapes, respectively (Table 7). While the first of these estimates is not beyond the realm of credibility for typical wood thrush densities (0.1-1.0 territories/ha; Roth et al. 1996), certainly the second estimate is biologically unreasonable. The behavior of this pair of models suggests that the assumption of model linearity over the range of simulated habitat conditions is unreasonable and that prediction of wood thrush density in hardwood stands based on observed vegetation characteristics in neighboring pine stands is suspect.

The selection of optimal burning and compartment permutation design alternatives was sensitive to model choice for the wood thrush response (Table 9). For most models (all models of classes $M_{0\cdot\cdot}$ and $M_{1\cdot\cdot}$), infrequent burning was clearly superior to frequent burning under any permutation alternative (Table 9). However, for models in the class $M_{2\cdot\cdot}$, the better alternatives did not always correspond to infrequent burning (Table 9). Under model M_{210}, the optimal decision called for frequent burning and high between-year and within-year compartment dispersion (Table 9). Under model M_{211}, the worst-ranked alternative corresponded to the Refuge's operational permutation sequence combined with infrequent burning (Table 9).

In light of the evident absence of wood thrush response variation with respect to permutation design (Table 6), apparent model-specific differences in decision alternative ranks within levels of burning could be due to chance (Table 9). Nevertheless, most of the apparently optimal permutation designs within each level of burning were those with high within-year compartment dispersion (9 of 12 models for infrequent burning, 10 of 12 for frequent burning; Table 9).

Thus, optimal management for wood thrush may in fact be model dependent, at least with regard to forest overstory model alternatives. I found no indication that optimal wood thrush management depended on choice of the habitat-association submodel.

The Refuge's operational permutation sequence was globally top-ranked in one case (model M_{011}) and bottom-ranked in five cases (models M_{000}, M_{010}, M_{110}, M_{111}, M_{211}) (Table 9). Within each level of burning, the operational permutation sequence was inferior to the middle-ranked alternative in eight of the twelve models (Table 9).

Under model uncertainty for the wood thrush response, any decision alternative in which burning was conducted infrequently was superior to alternatives in which burning was frequent (ranks for model-averaged outcomes, Table 9). Within levels of burning, permutation designs that featured high within-year compartment dispersion were generally superior to those that did not (Table 9). The Refuge's operational permutation sequence performed poorly both within each level of burning and overall (Table 9).

Composite Species Response

Despite the equal weighting given to the standardized woodpecker and wood thrush components of the composite species score, the composite score responded to management in ways similar to the woodpecker response, at least with regard to periodicity of burning. More frequent burning brought about greater mean responses in the composite species score, though burning effect sizes were moderate relative to those for the woodpecker response ($P \leq 0.144$, Tables 6-7).

With regard to compartment permutation design, patterns in variation for the composite species response showed characteristics expressed by one or the other of the constituent responses, and the patterns appeared model-related. For models in the class $M_{0\bullet\bullet}$, none of the successive-year, within-year, or interaction dispersion effects were of significance ($P \geq 0.124$, Table 6), but the pattern of effect means resembled that for the woodpecker response (Table 7). The composite mean response was positively associated with a combination of low successive-year and high within-year compartment dispersion, but the association was less distinct as it was for the woodpecker response (Table 7).

For models in classes $M_{1\bullet\bullet}$ and $M_{2\bullet\bullet}$, main effects for permutation design were significant only in isolated cases (Table 6). However, a consistent pattern emerged among the effect means. In all models, the composite species response was greater when within-year dispersion was high rather than small (Table 7). The successive-year dispersion effect expressed a pattern dependent on choice of the wood thrush habitat relationship submodel. The mean response was generally greater when successive-year dispersion was high rather than low in the linear forms of the model (models $M_{1\bullet 0}$ and $M_{2\bullet 0}$) and when dispersion was low rather than high in the nonlinear forms (models $M_{1\bullet 1}$ and $M_{2\bullet 1}$) (Table 7).

Composite response mean varied by submodel type. The composite response mean was greatest for models in class $M_{2\bullet\bullet}$ ($\bar{x} = 21.94$, SE = 0.0229), least for models in class $M_{0\bullet\bullet}$ ($\bar{x} = 16.69$, SE = 0.0116), and intermediate for models in class $M_{1\bullet\bullet}$ ($\bar{x} = 19.60$, SE = 0.0130). The mean for the habitat-sensitive versions of the woodpecker productivity model (class $M_{\bullet 1\bullet}$; $\bar{x} = 19.92$, SE = 0.0121) was greater than that for the alternative submodel (class $M_{\bullet 0\bullet}$; $\bar{x} = 18.89$, SE = 0.0149). The mean for the nonlinear forms of the wood thrush habitat-association model (class $M_{\bullet\bullet 1}$; $\bar{x} = 29.76$, SE = 0.0135) was far greater than that for the alternative submodel (class $M_{\bullet\bullet 0}$; $\bar{x} = 9.050$, SE = 0.0136), a result explained by the great difference in variation of wood thrush outcomes produced by the two classes of model.

The optimal decision alternative for the composite response depended on model choice (Table 10). Under all models, the optimal decision alternative incorporated frequent burning, whereas the bottom-ranked alternative employed infrequent burning. However, the form of the overstory transition model affected the clarity of the segregation of superior from inferior decisions. For most models in the classes $M_{0\bullet\bullet}$ and $M_{2\bullet\bullet}$, the five top-ranked decision alternatives corresponded to frequent burning, whereas for models in class $M_{1\bullet\bullet}$, some of the poorer alternatives used frequent burning (Table 10).

Within the high level of burning, the optimal decision alternative incorporated high within-year compartment dispersion in ten of the twelve models (Table 10). Of these ten, eight also employed low successive-year compartment dispersion (Table 10). Over all twelve models, the low-high combination of successive/within-year dispersion under frequent burning was never worse than the third-ranked decision alternative. Under the low level of burning, no particular expression of compartment dispersion conferred a clear advantage to the composite response (Table 10). The pattern of optimal permutation decisions under high burning frequency was almost identical to that for the woodpecker response (Table 8), whereas the pattern under low burning frequency closely resembled that for the wood thrush response (Table 9).

The Refuge’s operational compartment permutation sequence, when applied under a regime of infrequent burning, was the poorest decision alternative, or very nearly so, for eleven of the twelve models (Table 10). Under more frequent burning, the operational sequence was the second-highest ranked alternative for five of the models but fared relatively poorly (no higher than fourth-ranked) for the other seven (Table 10).

Under model uncertainty, the optimal decision alternative for the composite response was one in which burning was frequent, successive-year compartment dispersion was low, and within-year dispersion was high (ranks of model-averaged outcomes, Table 10). Within levels of burning, superior decision alternatives tended to be those having high within-year compartment dispersion, and the Refuge's operational permutation sequence was among the poorest alternatives (Table 10). Under frequent burning, the ranking of decisions for the composite response was identical to that for the woodpecker response (Table 8); under infrequent burning, the decision ranks were almost identical to those for the wood thrush response (Table 9).

ESTIMATION OF VALUE OF INFORMATION

The central notion of adaptive resource management is that while uncertainty about the response of a system to management is a primary impediment to its optimal management for a desired objective, management may nonetheless be directed, or at least evaluated *a posteriori*, toward the resolution of this uncertainty (Walters 1986). This notion brings forth at least two important implications that concern my simulation of management decision making at the Refuge. The first is that uncertainty about the system must be portrayed in a set of models that induce distinct model-dependent patterns of response to system management (Walters 1986:169, Johnson et al. 1993). If they do not, then resolution of model uncertainty is irrelevant as any model could be arbitrarily chosen to find an optimal decision action. If they do, then implicitly there exists a "value of information," expressible in units of the managed resource (Lindley 1985:120, Walters 1986:197, Dakins 1999). Value of information is the gain in the resource expected by reducing system uncertainty. Secondly, there must exist a source of data, for example, an ongoing monitoring program, and a means to apply these data to the reduction of uncertainty (Johnson et al. 1993).

I found that mean responses varied among uncertain models, as indicated by model-specific sets of outcome rankings (Tables 8-10). However, mean responses were stochastic, implying that different runs of the same models could produce different outcome rankings. Thus, some disagreements among models in decision alternative ranks are less meaningful than others, and in these cases, resolution of model uncertainty is not expected to be as helpful as may be apparent. For example, consider the optimal red-cockaded woodpecker response among models (Table 8). The top-ranked decision under model M_{100} was the fifth-ranked alternative under model M_{000}. However, the optimal objective value under model M_{100} was not statistically distinct from that of the second-ranked alternative, which was the optimal decision alternative in the majority (10) of the other models compared. Thus, in a practical sense, the optimal decision alternative was the same in eleven of the twelve models, and resolution of model uncertainty for this response may be an irrelevant exercise.

Values of information (Lindley 1985:120) computed from the simulation outcome means of each of the uncertain models (Tables 8-10) were 0.73 for the woodpecker cluster response, 0.006 for the wood thrush response, and 0.013 for the composite response. These values were small in comparison to corresponding average standard error values for the mean outcomes (Tables 8-10). Thus, when judged in the light of response precision, values of information may be negligible for these responses. On the other hand, because value of information is a measure derived from quantities subject to sampling variability, the measure itself has variability and an associated confidence interval, which may span a considerable range.

MODEL VALIDATION AND MODEL UPDATING

Prescribed burning was the only significant bird habitat management action occurring on the Refuge between the bird breeding seasons of 2000 and 2001 (J. A. Mason, Piedmont National Wildlife Refuge, personal communication). In the model, I updated the age-at-last-burn variable for those pine stands burned in the winter of 2000-2001. No stand-level thinning or regeneration took place in this period; managers carried out only isolated cuttings in response to southern pine beetle (*Dendroctonus frontalis*) infestations. No new recruitment clusters were installed in this time period.

Thirty-nine active red-cockaded woodpecker clusters were detected in sixteen compartments in 2001 (range 1-6 active clusters per occupied compartment, Table 11). Each model generally over-predicted cluster abundance at both the compartment and Refuge (range 42.6-42.9 mean active clusters) levels. The models all predicted some probability of occupation in 23 compartments (range 0.35-5.99 mean active clusters per occupied compartment, Table 11).

Conditional likelihood for the woodpecker response was greatest for model M_{111} and least for model M_{100} (Table 12). Corresponding posterior probabilities (p_1) for these models were 0.0896 and 0.0771, respectively (Table 12). Totaling p_1 over the wood thrush submodels yielded posterior probabilities 0.163, 0.169, 0.165, 0.175, 0.163, and 0.164 for model classes $M_{00\bullet}$, $M_{01\bullet}$, $M_{10\bullet}$, $M_{11\bullet}$, $M_{20\bullet}$, and $M_{21\bullet}$, respectively (compare to equal prior probabilities of 0.167). Thus, based on the woodpecker cluster response, evidence accumulated toward the model classes $M_{01\bullet}$ and $M_{11\bullet}$ and diminished from the others. These two classes differed in the rate of transition to hardwood, but they both proposed that woodpecker recruitment was sensitive to foraging habitat abundance.

Posterior probabilities totaled over both the woodpecker and wood thrush submodels were 0.332, 0.340, and 0.328 for model classes $M_{0\cdot\cdot}$, $M_{1\cdot\cdot}$, and $M_{3\cdot\cdot}$, respectively (compare to equal prior probabilities of 0.333). Thus, the woodpecker cluster response was more consistent with the hypothesis of aggressive hardwood encroachment than with the hypothesis of old-growth pine retention. Total posterior probabilities with respect to the woodpecker submodels were 0.491 for class $M_{\cdot 0\cdot}$ and 0.509 for class $M_{\cdot 1\cdot}$ (compare to equal prior probabilities of 0.500). Therefore, models that proposed habitat-sensitive woodpecker recruitment received a slightly larger share of credibility weight following data collection.

CHAPTER 6

DISCUSSION

SYNTHESIS OF MODELING RESULTS

Populations of red-cockaded woodpeckers and wood thrushes, as expressed through the set of simulation models, were sensitive to the decision alternatives. For both populations, annual extent of burning greatly influenced the predicted 100-year abundance of birds. Woodpecker populations responded positively to increased burning, whereas wood thrushes responded negatively. The specific configuration of compartments into management groups was less influential on mean outcomes. In general, the woodpecker population received the greatest benefit from management when compartments within management groups were widely dispersed and when compartments in successive-year groupings were close to each other. Wood thrushes generally benefitted under management group configurations that maximized within-year compartment dispersion. Optimal decision actions for the composite species score more closely resembled those for the woodpecker response than those for the wood thrush response.

Likewise, predicted abundances of both woodpeckers and wood thrushes were sensitive to choice of system dynamics model. Abundance of active woodpecker clusters responded strongly to choice of forest overstory submodel and less strongly to choice of woodpecker productivity submodel. Choices of forest overstory and wood thrush habitat-association submodels both substantially influenced wood thrush mean outcome.

Despite the unquestionable influence of both decision action and model choice on mean outcome, the selection of the optimal decision action for each response appeared mostly insensitive to model choice. Even for the case of wood thrush, where decision outcome rankings were most variable among models, superiority of decisions was not clearly evident as differences in decision outcomes were not large relative to outcome variability. Furthermore, estimated values of information were extremely low relative to outcome variances, suggesting that little management value is to be gained by resolving uncertainty among these models.

The models all over-predicted abundance of active woodpecker clusters in 2001. Model-specific outcomes for a single-year time step were highly consistent with each another. However, some models performed slightly better than others in projecting the year 2001 abundance of clusters. As a result, models which proposed aggressive hardwood encroachment and habitat-influenced woodpecker recruitment received greater credibility weight after confrontation with data than those that did not. Re-computation of model-averaged outcomes through application of the posterior probabilities (Table 12) to model-specific decision outcomes (Tables 8-10) did not change the decision ranking for the woodpecker response (Table 8). However, the second and third-ranked decision alternatives for the composite species response under model uncertainty (Table 10) swapped rankings following updating of the model credibility weights.

The updating of the credibility weights described above is an example of passive adaptation. Furthermore, my simulation and optimization approach was an example of open-loop rather than closed-loop optimization (Dreyfus and Law 1977). That is, the approach assumed that the optimal decision constituted a fixed time series of decisions, rather than a single current-time decision anticipating change in the future system state. The optimal decision for a given system state under closed-loop optimization is not necessarily the same as that under open-loop optimization because the past history of the system is irrelevant in the former case (Markovian) but not in the latter case (non-Markovian). Closed-loop optimization requires dynamic programming methodology, which was clearly unable to accommodate this problem. Unfortunately, because open-loop approaches do not anticipate changes in the system state, closed-loop approaches may be the only ones capable of delivering actively adaptive decision policies.

IMPLICATIONS FOR REFUGE MANAGEMENT

The optimal decision under model uncertainty for a composite woodpecker and wood thrush population response called for extensive burning and compartment groupings having high within-group dispersion and low successive-group dispersion. This decision corresponded to equal weight provided to the variance-scaled responses for woodpecker and wood thrush. Allocating more weight to the wood thrush response would indicate a different optimal decision.

Relative to these actions, Refuge burning is limited in area, and current inter-compartment dispersion is moderate both within management groups and between successive groups. Whereas increasing the extent of burning may be difficult within current resource levels and operational constraints, management group configuration may be feasible to address.

I found little variation in the optimal decision among models, and I did not detect high values of information in the analysis. Nevertheless, one should not conclude that key uncertainties do not exist in this system nor that management cannot be used to improve quality of decisions. In fact, my work has revealed that this system is highly uncertain, and model-based decision making on the Refuge remains frustrated by four main obstacles: doubt over clarity and appropriateness of objectives, insufficient resource data at appropriate resolutions, lack of suitable predictive models, and unclear expression and questionable breadth of decision alternatives.

OBJECTIVES

The form of an optimal policy is determined by the objectives. I analyzed bird abundance outcomes separately over a 100-year time horizon using 20 iterations of each model for each decision alternative. My selection of this time frame and number of iterations was a compromise between the desire to obtain a consistent, long-term depiction of management results on the Refuge and the need to complete the simulations in a reasonable time period. The time frame was just sufficient to cover a single forest rotation. Given the extensive degree of hardwood encroachment evident in the simulations, a longer time frame would have suggested different decision outcomes. In the example set of simulations (Figure 18), abundance of active red-cockaded woodpecker clusters appeared to peak, then decline.

I also analyzed bird outcomes simultaneously through the use of a composite measure. By choosing this measure, I attempted to balance influences by the individual components in the selection of optimal decision alternatives. Though I gave each component equal weight in the composite score, precision scaling supplied greater influence to the woodpecker component than to the wood thrush component. Although it is reasonable and is common practice to apply an inverse-variance scaling when averaging components measured on highly dissimilar scales, the scaling quantity that I chose likely reflected differences in the quality or structure of the population models and their inputs rather than true differences in population variability.

A more troublesome issue is whether overall conservation planning for the Refuge is even properly accomplished through the maximization of abundance of two forest bird species. As discussed earlier (Chapter 2), conservation objectives for the Refuge are not totally clear-cut, given the desire to perpetuate the red-cockaded woodpecker population within the context of ecosystem management. If ecosystem management implies the use of management to mimic historical ecological processes and patterns, then the persistence of woodpeckers in this landscape, over any long-term basis, is a questionable goal for management. Instead, a management objective that seeks to optimize a broader but measurable index of biotic diversity may be more appropriate, as might be one that spatially separates species objectives.

Although I selected the wood thrush as a representative species of understory-dependent non-target fauna, how well it represents those fauna is unknown. Because the wood thrush migrates, inferences on forms of management that appear benign to the wood thrush certainly cannot be extended to terrestrial organisms. Within closed-canopy forest conditions, wood thrushes are to some degree habitat generalists. Therefore, forest management that appears beneficial or, at least, not harmful to wood thrushes may be highly destructive to a single species of plant.

RESOURCE DATA

Modeling and validation efforts for projecting forest response to management are presently hampered for lack of a systematic, comprehensive forest data collection program at the Refuge. Any forest decision-making endeavor, whether adaptive or not, must be based on a program of regular forest monitoring. Monitoring data provide the means by which progress toward objectives are measured, and, in an adaptive setting, they arbitrate among competing decision models. Furthermore, they can be used to develop or refine models.

The habitat inputs that I supplied to the models were measured at different extremes of spatial scale (approx. range 4-400 ha) and precision. For example, I initiated stand basal area in the GIS by calculating the midpoint of an estimated compartment-wide range of basal area then propagating the estimate into all stands. Such gross spatial inaccuracies bring into question the merit of any spatially-explicit modeling effort. Furthermore, no data exist to validate the alternative forest overstory models I developed. As all species responses and optimal regeneration decisions were highly sensitive to choice of forest overstory model, lacking the means to resolve uncertainty among these models is a serious impediment to improved decision making. Finally, records of management actions are maintained in paper form. These records are difficult to synthesize in a format useful for understanding the timing, distribution, and effects of historical decision actions.

To make management more informative, I strongly encourage the Refuge to initiate a point-based monitoring and data management system. Such a system would provide spatially-referenced data on site characteristics, forest conditions, management history, disturbances, and certain forest fauna. Presently, stand polygons are the basic mapping element on compartment maps. This representation of the forest is too coarse to be useful in the construction of spatially-explicit forest and animal population or community models. Instead, my modeling efforts point to the versatility and usefulness of a point-based monitoring program. A monitoring system established on a regular, fixed grid of sampling points would provide data in a form that is readily incorporated in a GIS and linkable to predictive models.

At each of 0.0405-ha (0.1-ac) permanent plots, conditions of the site (site index, topographic characteristics) and the overstory and midstory layers (composition, age, basal area, canopy closure and volume) would be collected. One-percent coverage of the Refuge would require approximately 3500 such plots. However, because these forest attributes change through time somewhat slowly (or not at all), all plots in a compartment could be sampled only when the compartment is visited for silvicultural treatment. Because the treatment cycle occurs over eight years, fewer than 500 plots would be surveyed for these attributes in any single year. Although this approach samples only a portion of the Refuge each year, a composite picture of forest conditions throughout the Refuge can be generated whenever desired. Forest growth models embedded in the database can advance forest attributes to any chosen point in time.

In contrast, dynamics of the understory occur rapidly, and decisions on burning are made annually on a Refuge-wide basis. Therefore, data on understory conditions should be collected each year on a random or systematic sample of plots throughout the Refuge. A sample of 700 plots provides approximately 0.2% Refuge coverage. The principal attributes collected in this survey are those depicting understory vegetation composition and density.

At each point affected by a management action, data on timing, type, and characteristics of the action should be recorded in the database immediately following the action. Important attributes to be recorded include the date and duration of the most recent cutting activity, the type of cut, residual basal area and canopy closure, year of pine establishment, date of most recent prescribed burn, and characteristics of the burn. Similarly, immediately following a forest disturbance, data would be recorded for each point affected by the disturbance. Recorded attributes include timing and type of disturbance (e.g., beetle damage, wind throw, etc.) and characteristics of the residual stand.

Separate protocols should be established for forest wildlife monitoring. However, many vertebrate and invertebrate monitoring designs can be overlaid on the forest sampling grid. For example, surveys of songbirds could be conducted on a habitat-stratified sample of the forest monitoring points. Currently, point counts of songbirds are collected annually, but the sample design is not stratified and the data collected do not yield songbird density. By recording distances to birds detected, estimates of songbird density can be made in each compartment and habitat stratum. For certain other species however, monitoring is best accomplished by restricting sampling to specific habitats (e.g., streams) or sites (e.g., woodpecker clusters).

Improvements in the models of woodpecker cluster dynamics will only occur with better woodpecker demographic data. Data currently collected are numbers of active woodpecker clusters, group size per cluster, and productivity per cluster. These data are collected annually at all clusters throughout the Refuge. However, woodpeckers are not banded. Thus, inferences on survival, pair fidelity, and dispersal are not available from the data.

Data provided by the proposed monitoring system could generate high-resolution GIS information layers and provide spatially relevant input to decision models. These data would be absolutely critical for assessing performance of competing decision models, redistributing belief weight among models, refining model structure and parameters, and providing a basis for the development of new models. Furthermore, the use of systematic surveys gives managers a means of mapping the biological diversity of the Refuge and tracking its response to management. As Refuge management continues to emphasize woodpecker recovery objectives, a systematic monitoring program is the best means by which to assess effects of management on non-target organisms.

PREDICTIVE MODELS

All analyses I performed were conditional on the model structure and parameter values that I chose, and I made many choices subjectively. The models were assembled from assorted component models – some components were developed from data collected on the Refuge while others were developed for situations elsewhere. Some of the component models were process-oriented (e.g., woodpecker productivity) while others were statistical (e.g., wood thrush abundance). A sensitivity analysis could reveal those parts of the simulation model most vulnerable to uncertain parametric controls, but this sort of analysis was not possible to perform under the extreme computing demands of the model. Thus, there is no assurance that the alternative model forms I used actually reflect the greatest uncertainties about forest and bird management in this system.

Recent modeling efforts for the red-cockaded woodpecker have emphasized spatially-explicit, individual-based approaches (Hughell 1996, Letcher et al. 1998, Walters et al. 2002), but such approaches involve the specification of a number of unknown parameters with uncertain implications for model reliability (Conroy et al. 1995, Ruckelshaus et al. 1997). Furthermore, such models must be linked to the landscape when spatially-distributed habitat management actions are of interest. Though these models are increasingly recognized as important tools for conservation management, they are difficult to develop, and few examples exist of their use in conservation management.

To my knowledge, there exists no management model that treats red-cockaded woodpecker population dynamics in a spatially-explicit, individual-based (or group-based) form and in response to local habitat conditions, yet this is the type of model believed to be required for recovery and conservation planning for the woodpecker (U.S. Fish and Wildlife Service 2000). The models I developed represent a first step toward this goal in that they contain mathematical structures that relate the life processes of the woodpecker family group to the habitat, the population, and the stochastic environment. Though some parameter values were derived from study of the Refuge population, others were merely best guesses.

Models for forest overstory dynamics were based mostly on supposition about habitat succession processes and very little on empirical data, whereas models for wood thrush abundances were based entirely on observed correlations and not at all on population dynamics. These approaches may have been reasonable in light of the data available, but certainly others have equal merit. To produce a set of models likely to encompass the principal uncertainties in the bird conservation community, a collaboration among resource biologists is almost compulsory.

Because many model parameters were set either subjectively or on the basis of off-site empirical data, these models must be considered provisional until data on the Refuge either confirm these values or indicate better ones. Thus, data obtained through a Refuge-wide systematic monitoring program are vital to the improvement of these models and the development of new models.

The models I used are highly detailed and contain many parameters. This is a consequence of my decision to employ spatially-explicit, breeding-group-based population models on a heterogeneous, dynamic landscape. An obvious question is whether such an approach is unnecessarily complicated and whether some simplification of decision models is justified. Clearly, sensitivity analysis could address this question, but this was not feasible for the landscape simulation model. Because woodpecker dispersal is so limited and because woodpeckers are so highly restricted to an ephemeral habitat, I concur with the view that management purely for woodpecker objectives should proceed in a spatially-explicit, stand-level, and individual-based context (U.S. Fish and Wildlife Service 2000). This may especially be the case in the Piedmont and in other non-longleaf pine habitats outside of the coastal plain. In terms of physiographic, edaphic, and forest cover attributes, these areas tend to be more spatially heterogeneous than the coastal plain, and maintaining a stream of suitable habitat in appropriate spatial configurations is relatively difficult to assure.

However, simplification may be warranted where forest biological diversity is the objective of management. In this case, landscape attributes that are controllable by management activities may be related to diversity outcomes (e.g., total species in an area, number of rare species) through relatively simple, correlative, non-dynamic models. Thus, whereas the abundance and distribution of old pine habitat may be predicted in a landscape model, woodpecker use of that habitat may be predicted only in probabilistic terms.

In the case of forest overstory transition, a simple model seemed reasonable to apply, and this model provided great insight into regeneration planning on the Refuge. Under any scenario of hardwood encroachment under this model, optimal pine harvest amount (252-901 ha) for an old-growth pine objective far exceeded the average amount (54 ha) that the Refuge is currently harvesting. The model alternatives project greatly different overstory composition trajectories through time. Although it is not known which of these alternatives most closely approximates truth, the facts that the alternatives are so different, that they imply very different optimal harvest decisions, and that these decisions all exceed current regeneration quantities provide some indication that the current rate of regeneration is, in fact, suboptimal for providing woodpecker nesting habitat.

The divergent approaches that I took in the modeling of woodpecker and wood thrush population responses suggest three levels in the hierarchy of structural uncertainty: uncertainty with regard to the class of model (e.g., population dynamics vs. habitat-correlative), uncertainty in the mathematical structure of a model, and parametric uncertainty. However, these distinctions are important only to the extent that they indicate differential responses to management (Walters 1986). I was not able to easily explore alternative model classes in my work, in part because I lacked the critical data needed to build and evaluate such models.

As stated earlier, information about the relative merit of each alternative model is not available without a concurrent program of systematic Refuge monitoring. With the exception of woodpecker monitoring, no feature of the forest is currently measured that allows the updating of model information, at least not at the resolution or in the metric consistent with model output. Therefore, the Refuge would benefit from a program of monitoring (see above) that informs managers about the results of their actions.

I combined the separate forest dynamics, woodpecker, and wood thrush component models into twelve distinct simulation models, and I used the 2001 woodpecker data to infer quality of each combination. A better approach would be to evaluate the components separately using corresponding sets of data. For example, spatial data collected on forest overstory composition could inform the manager on the relative quality of each of the three forest submodels. Furthermore, despite the fact that the monitoring program I propose only samples the overstory in one-eighth (approximately) of the compartments each year, the revolving protocol for sampling would nevertheless provide for the annual updating of forest models.

Decision Alternatives

Especially challenging was establishing and modeling the set of decision alternatives. Decisions had to permit spatial flexibility, yet belong to a sufficiently limited set to facilitate simulation over a long time frame. Decisions had to be operationally feasible such that all actions called for in a simulation year could actually be implemented by Refuge managers. Decisions also had to conform to restrictions imposed by legal mandates and species recovery plans.

The set of decision alternatives I used may have been too narrow to meaningfully address uncertainty. The few combinations of compartment permutation and burn periodicity that I selected were only a fraction of the range of available decision alternatives. Furthermore, this total range of alternatives was itself quite narrow because I restricted my focus to compartment permutation alternatives of the type that is currently employed by the Refuge. In addition, I enforced a rule-based set of stand-level actions in conformance with guidelines in red-cockaded woodpecker recovery plan documents (U.S. Fish and Wildlife Service 1998, 2000).

Part of the difficulty in establishing decision options was due to the range of spatial and temporal scales of the actions and the inconsistency of these scales with resource measurement scales. For example, decisions on regeneration and thinning are made both at the management group (multi-compartment) and stand scales at eight-year intervals. After a compartment is treated, it is not reevaluated for further cutting treatment (excluding treatments for insect damages and other localized disturbances) for eight years. In contrast, woodpeckers are counted annually at the cluster scale and wood thrushes are monitored at the compartment scale. Thus information extracted from the system is available at relatively fine scales of time (year) and space (compartment), but it is unclear how to effect a mid-course correction in the management group rotation sequence in response to the collected information. One alternative is to simply reevaluate the question of compartment visitation each year. That is, a new optimal compartment configuration is estimated and acted upon as if it will be in place for eight years.

The analysis indicated that the woodpecker population was most responsive to a particular schedule of compartment silvicultural treatments. Perhaps by fixing this schedule in the simulation model, other sets of decision alternatives could be considered and explored more fully. For example, given the high degree of habitat heterogeneity and biotic diversity of Refuge, decision alternatives relating to type of silvicultural system (e.g., even-aged versus uneven-aged management) or aggressiveness of hardwood midstory reduction (e.g., dormant versus growing-season burning, mechanical removal, or combinations of burning and mechanical removal) may be more appropriate avenues of investigation than the ones I explored.

ADAPTIVE MANAGEMENT IN LANDSCAPE-LEVEL CONSERVATION PROBLEMS

My work attempted to investigate and demonstrate how principles of adaptive management can be applied to conservation management in a landscape setting. There are clear advantages to using adaptive approaches in conservation management, though there also exist many challenges in their application.

Adaptive management allows decision making to proceed in the data-poor environments that often characterize conservation problems. The key is the development of alternative plausible system models that capture the breadth of system uncertainty. Each model admits a competing view on how populations respond to management, and each implies a different optimal course of management. Though data may be helpful in constructing the models, they are not absolutely necessary. Thus, with a robust set of decision models, where each model is developed conceptually rather than empirically, an optimal course of management can be derived even in the absence of data.

However, regardless of the availability of data for model development, adaptive management requires a commitment to resource monitoring. The collection of data on the managed system is the means by which the competing models are given either greater or less influence in future decisions. Data should be collected on all resource components for which competing models generate predictions. For the Refuge case, these components would be forest overstory composition, woodpecker cluster status, and wood thrush abundance.

Controversy that arises due to uncertainty in the management of a resource can be controlled and directed through adaptive management. Because adaptive management requires formal statements of management objectives and explicit models of system response to management, disagreements over management objectives are decoupled from uncertainties about responses. Any stakeholder in the decision is eligible to submit a system model that expresses their belief in the behavior of the system. Monitoring data are then used to arbitrate among the competing models. Therefore, disagreements in the objectives of management, which may be vigorous and may require painful negotiation among parties, cannot be disguised in unwinnable arguments over whose version of science and favored management policy is “right.”

Defining the objectives of conservation management is often not straightforward. I chose long-term abundances of two species as objectives in this study. Whether or not these objectives are appropriate for Refuge management is open to debate, but such objectives are probably overly simplistic in some management settings. In most conservation settings, a broader diversity objective is likely to be favored, but diversity objectives cannot be entertained without also making a commitment to a program of system monitoring. Managers and policy makers face challenges in defining the future states they would like to achieve, keeping these goals explicit, and assuring that these goals do not change with each decision cycle.

Collaboration is practically a necessity for successful application of adaptive management. Conservation biologists and other qualified decision stakeholders may best be able to summarize the scientific literature, as well as express their own perceptions, convictions, and biases, in a set of competing models. Resource managers are best suited for taking the lead in developing the set of decision alternatives and in devising a monitoring program. Policy makers have an important role in collecting input from all decision stakeholders and developing a clear statement of management objectives. Team members should be able to contribute skills in statistics, monitoring design, computer science, human dimensions study, operations research, GIS and database architecture, and population modeling.

Efforts to make management models and decision alternatives increasingly realistic, for example, use of high spatial resolution, multiple stochastic elements, or elaborate demographic structure, will be met by greater computational challenges in their analysis. Until substantial increases in computing power or innovations in computational technique are realized, closed-loop, active adaptive optimization may be unavailable for all but the most simplistically-formulated conservation problems. Even so, simulation-based approaches may compare favorably to bona fide optimization (Moore et al. 2000, Conroy and Moore 2001; Appendix E).

Sets of competing decision models, monitoring programs for data feedback, explicit statements of objectives, and interdisciplinary collaboration are elements not often found in the decision making toolboxes of many resource management agencies. Yet these tools are necessary for successful implementation of adaptive management. If adaptive management is to become reality in resource agencies, agencies must commit to adopting such tools and to creating a genuine environment in which scientific uncertainty is admitted and confronted.

CHAPTER 7

CONCLUSION

My work demonstrated the application of adaptive management principles to a problem in landscape management where the resources of interest are dynamic, sensitive to the spatial distribution of management actions, and differentially responsive to a single set of actions. Each of the tools of adaptive management found use in my application: (1) a set of competing, predictive system models, (2) an explicit statement of management objectives, (3) a list of decision alternatives, (4) a program of monitoring responses to management, and (5) assignment of credibility measures to each competing model and a means of updating these measures through the confrontation of model predictions to resource data.

I found that extensive use of fire and allocation of forest cutting actions (thinning and regeneration) in a particular type of spatial sequence over time was most beneficial for a composite measure of response by populations of red-cockaded woodpecker and wood thrush. This finding was mostly consistent over all model scenarios. Thus, the resolution of uncertainty with regard to the system models that I used would not be profitable in terms of improved management performance.

For several reasons, however, these results should be considered provisional. First, the objectives I investigated, long-term abundances of active woodpecker clusters and wood thrushes, may not best reflect a forest diversity response, if that is the true goal of Refuge management. Second, most model parameters either were chosen subjectively or were based on data collected off-site, therefore, applicability of the models in this system are tentative until model parameters can be investigated for sensitivity or until they can be estimated from data obtained at the Refuge. Third, the greater influence of the woodpecker response in the composite species response could have been an artifact of my use of completely different forms of population model for the woodpecker and wood thrush response. Fourth, the set of decision alternatives was somewhat narrowly defined by institutional, logistical, and computational constraints. In particular, the response to alternative silvicultural systems would be a worthwhile topic of study, but it was not addressed in my work.

Finally, the Refuge lacks a program of systematic forest monitoring and data storage. Establishing such a program is a matter of high priority if management is to be informative. A forest monitoring program would supply a baseline of data on forest diversity, and effects of management on diversity can be tracked. A monitoring program would provide information for improving decision models and for evaluating performance among models. The monitoring program would also preserve data on management history performed throughout the Refuge so that the link between action and response can be better understood. My decision modeling led to specific recommendations for the conduct of system monitoring at the Refuge.

Adaptive management serves as a means of making resource management more transparent among managers, scientists, policy makers, and the public. The explicit steps in setting objectives, listing decision alternatives, and forecasting possible system responses to each action bring openness and a measure of discipline to the decision-making process, a process that often involves multiple stakeholders. Such a degree of transparency in decision making is not customarily encountered in traditional approaches to natural resource management. Despite the many technical challenges involved in implementing a model-based approach to decision making, certainly such an approach is preferable to one in which a decision maker does not or cannot describe his beliefs and personal biases about the effects of management on the system (i.e., the system is treated as a "black box"), disagreeing stakeholders cannot express how their opinions differ regarding system response to management, or in which the definition of a "successful" management decision is left to the subjective judgment of the decision maker. Adaptive management provides a framework that exposes each of these elements to critical examination and public discussion.

In the new environment of ecosystem management, natural resource agencies are placing greater emphasis on managing for ecosystem-wide structure, services, and processes and less emphasis on the return of single-resource commodities (Christensen et al. 1996, Keiter 1998, Malone 2000). Grumbine (1994) describes ecosystem management as a participatory framework for the application of scientific knowledge to the making of management decisions for diverse ecological and social objectives. Thus, adaptive management serves as a natural vehicle for achieving the goals of ecosystem management (Grumbine 1994, Christensen et al. 1996, Jensen et al. 1996). Predictions of ecosystem response to management actions and stochastic events are only available through system models. Understanding of ecosystem processes is increased only through the collection of data, and this increased understanding is expressed through improved models. Finally, avenues for participation by scientists, managers, and stakeholders occur throughout the steps of forming objectives, developing models, and monitoring the system. For these reasons, it is hard to imagine progress toward ecosystem goals in conservation management without an adaptive approach to decision making.

LITERATURE CITED

Akaike, H. 1973. Information theory as an extension of the maximum likelihood principle. Pages 267-281 *in* B. N. Petrov and F. Csáki, eds. *Proceedings of the second international symposium on information theory*. Akadémiai Kiadi, Budapest, Hungary.

Anderson, D. R. 1975. Optimal exploitation strategies for an animal population in a Markovian environment: a theory and an example. *Ecology* 56:1281-1297.

Annand, E. M., and F. R. Thompson, III. 1997. Forest bird response to regeneration practices in central hardwood forests. *Journal of Wildlife Management* 61:159-171.

Azevedo, J. C. M., S. B. Jack, R. N. Coulson, and D. F. Wunneburger. 2000. Functional heterogeneity of forest landscapes and the distribution and abundance of the red-cockaded woodpecker. *Forest Ecology and Management* 127:271-283.

Bailey, R. L., and K. D. Ware. 1983. Compatible basal-area growth and yield model for thinned and unthinned stands. *Canadian Journal of Forest Research* 13:563-571.

Bart, J., and E. D. Forsman. 1992. Dependence of northern spotted owls *Strix occidentalis caurina* on old-growth forests in the western USA. *Biological Conservation* 62:95-100.

Bechtold, W. A., G. A. Ruark, and F. T. Lloyd. 1991. Changing stand structure and regional growth reductions in Georgia's natural pine stands. *Forest Science* 37:703-717.

Bellman, R. E. 1957. *Dynamic programming*. Princeton University Press, Princeton, New Jersey, USA.

Bertin, R. I. 1977. Breeding habitats of the Wood Thrush and Veery. *Condor* 79:303-311.

Bettinger, P., J. Sessions, and K. Boston. 1997. Using Tabu search to schedule timber harvests subject to spatial wildlife goals for big game. *Ecological Modelling* 94:111-123.

Beyer, D. E., Jr., R. Costa, R. G. Hooper, and C. A. Hess. 1996. Habitat quality and reproduction of red-cockaded woodpecker groups in Florida. *Journal of Wildlife Management* 60:826-835.

Block, W. M., D. M. Finch, and L. A. Brennan. 1995. Single-species versus multiple-species approaches for management. Pages 461-476 *in* T. E. Martin and D. M. Finch, editors. *Ecology and management of Neotropical migratory birds*. Oxford University Press, New York, New York, USA.

Bodansky, D. 1991. Scientific uncertainty and the precautionary principle. *Environment* 33(7):4-5,43-44.

Brackbill, H. 1958. Nesting behavior of the Wood Thrush. *Wilson Bulletin* 70:70-89.

Brawn, J. D., and S. K. Robinson. 1996. Source-sink population dynamics may complicate the interpretation of long-term census data. *Ecology* 77:3-12.

Brender, E. V. 1974. Impact of past land use on the lower Piedmont forest. *Journal of Forestry* 72:34-36.

Brennan, L. A., J. L. Cooper, K. E. Lucas, B. D. Leopold, and G. A Hurst. 1995. Assessing the influence of red-cockaded woodpecker colony site management on non-target forest vertebrates in loblolly pine forests of Mississippi: study design and preliminary results. Pages 309-319 *in* D. L. Kulhavy, R. G. Hooper, and R. Costa, editors. *Red-cockaded woodpecker: recovery, ecology and management*. Center for Applied Studies in Forestry, Stephen F. Austin State University, Nacogdoches, Texas, USA.

Brown, R. S., and K. Marshall. 1996. Ecosystem management in state governments. *Ecological Applications* 6:721-723.

Buckland, S. T., D. R. Anderson, K. P. Burnham, and J. L. Laake. 1993. *Distance sampling: estimation of biological populations*. Chapman and Hall, London, UK.

Buckland, S. T., and D. A. Elston. 1993. Empirical models for the spatial distribution of wildlife. *Journal of Applied Ecology* 30:478-495.

Burger, L. W., Jr., C. Hardy, and J. Bein. 1998. Effects of prescribed fire and midstory removal on breeding bird communities in mixed pine-hardwood ecosystems of southern Mississippi. *Tall Timbers Fire Ecology Conference Proceedings* 20:107-113.

Burnham, K. P., and D. R. Anderson. 1998. *Model selection and inference*. Springer-Verlag, New York, New York, USA.

Caldwell, L. K., C. F. Wilkinson, and M. A. Shannon. 1994. Making ecosystem policy: three decades of change. *Journal of Forestry* 92(4):7-10.

Carey, A. B., and R. O. Curtis. 1996. Conservation of biodiversity: a useful paradigm for forest ecosystem management. *Wildlife Society Bulletin* 24:610-620.

Caswell, H. 2001. *Matrix population models*. Second edition. Sinauer Associates, Sunderland, Massachusetts, USA.

Chadwick, N. L., D. R. Progulske, and J. T. Finn. 1986. Effects of fuelwood cutting on birds in southern New England. *Journal of Wildlife Management* 50:398-405.

Chamberlin, T. C. 1890. The method of multiple working hypotheses. *Science* (old series) 15:92. [reprinted 1965 in *Science* 148:754-759]

Charles, A. T. 1998. Living with uncertainty in fisheries: analytical methods, management priorities and the Canadian groundfishery experience. *Fisheries Research* 37:37-50.

Christensen, N. L., A. M. Bartuska, J. H. Brown, S. Carpenter, C. D'Antonio, R. Francis, J. F. Franklin, J. A MacMahon, R. F. Noss, D. J. Parsons, C. H. Peterson, M. G. Turner, and R. G. Woodmansee. 1996. The report of the Ecological Society of America Committee on the Scientific Basis for Ecosystem Management. *Ecological Applications* 6:665-691.

Collie, J. S., and C. J. Walters. 1993. Models that 'learn' to distinguish among alternative hypotheses. *Fisheries Research* 18:259-275.

Congressional Research Service. 1994. Ecosystem management: federal agency activities. Congressional Research Service, Library of Congress, Washington, D.C., USA.

Conner, R. N., and C. S. Adkisson. 1975. Effects of clearcutting on the diversity of breeding birds. *Journal of Forestry* 73:781-785.

Conner, R. N., and D. C. Rudolph. 1989. Red-cockaded woodpecker colony status and trends on the Angelina, Davy Crockett, and Sabine National Forests. USDA Forest Service Research Paper SO-250.

Conner, R. N., and D. C. Rudolph. 1991. Forest habitat loss, fragmentation, and Red-cockaded Woodpecker populations. *Wilson Bulletin* 103:446-457.

Conner, R. N., D. C. Rudolph, R. R. Schaefer, D. Saenz, and C. E. Shackelford. 1999. Relationships among red-cockaded woodpecker group density, nestling provisioning rates, and habitat. *Wilson Bulletin* 111:494-498.

Conner, R. N., A. E. Snow, and K. A. O'Halloran. 1991. Red-cockaded woodpecker use of seed-tree/shelterwood cuts in eastern Texas. *Wildlife Society Bulletin* 19:67-73.

Conroy, M. J. 1993. The use of models in natural resource management: prediction, not prescription. *Transactions of the North American Wildlife and Natural Resources Conference* 58:509-519.

Conroy, M. J., Y. Cohen, F. C. James, Y. G. Matsinos, and B. A. Maurer. 1995. Parameter estimation, reliability, and model improvement for spatially explicit models of animal populations. *Ecological Applications* 5:17-19.

Conroy, M. J., and C. T. Moore. 2001. Simulation models and optimal decision making in natural resource management. Pages 91-104 *in* T. M. Shenk and A. B. Franklin, editors. *Modeling in natural resource management*. Island Press, Washington, D.C., USA.

Conroy, M. J., and C. T. Moore. 2002. Wildlife habitat modeling in an adaptive framework: the role of alternative models. Pages 205-218 *in* J. M. Scott, P. J. Heglund, M. L. Morrison, M. Raphael, J. Haufler, and B. Wall, editors. *Predicting species occurrences: issues of accuracy and scale*. Island Press, Covelo, California, USA.

Conroy, M. J., and B. R. Noon. 1996. Mapping of species richness for conservation of biological diversity: conceptual and methodological issues. *Ecological Applications* 6:763-773.

Conroy, M. J., R. G. Oderwald, and T. L. Sharik. 1982. Forage production and nutrient concentrations in thinned loblolly pine plantations. *Journal of Wildlife Management* 46:719-727.

Copeyon, C. K. 1990. A technique for constructing cavities for the red-cockaded woodpecker. *Wildlife Society Bulletin* 18:303-311.

Costa, R. 1995. Outlook for recovery of the red-cockaded woodpecker. Pages 3-5 *in* D. L. Kulhavy, R. G. Hooper, and R. Costa, editors. *Red-cockaded woodpecker: recovery, ecology and management*. Center for Applied Studies in Forestry, Stephen F. Austin State University, Nacogdoches, Texas, USA.

Crawford, H. S., R. G. Hooper, and R. W. Titterington. 1981. Songbird population response to silvicultural practices in central Appalachian hardwoods. *Journal of Wildlife Management* 45:680-692.

Czuhai, E., and C. T. Cushwa. 1968. A resume of prescribed burnings on the Piedmont National Wildlife Refuge. USDA Forest Service Research Note SE-86.

Dakins, M. E. 1999. The value of the value of information. *Human and Ecological Risk Assessment* 5:281-289.

Davenport, D. E., R. A. Lancia, J. R. Walters, and P. D. Doerr. 2000. Red-cockaded woodpeckers: a relationship between reproductive fitness and habitat in the North Carolina Sandhills. *Wildlife Society Bulletin* 28:426-434.

DeGraaf, R. M., J. B. Hestbeck, and M. Yamasaki. 1998. Associations between breeding bird abundance and stand structure in the White Mountains, New Hampshire and Maine, USA. *Forest Ecology and Management* 103:217-233.

DeLotelle, R. S., and R. J. Epting. 1988. Selection of old trees for cavity excavation by red-cockaded woodpeckers. *Wildlife Society Bulletin* 16:48-52.

Dettmers, R., and J. Bart. 1999. A GIS modeling method applied to predicting forest songbird habitat. *Ecological Applications* 9:152-163.

Dickson, J. G. 1981. Effects of forest burning on songbirds. Pages 67-72 *in* G. W. Wood, editor. *Prescribed fire and wildlife in southern forests*. Belle W. Baruch Forest Science Institute, Georgetown, S.C., USA.

Dickson, J. G., F. R. Thompson, III, R. N. Conner, and K. E. Franzreb. 1993. Effects of silviculture on neotropical migratory birds in central and southeastern oak pine forests. Pages 374-385 *in* D. M. Finch and P. W. Stangel, editors. *Status and management of neotropical migratory birds*. USDA Forest Service General Technical Report RM-229.

Doak, D. 1989. Spotted Owls and old growth logging in the Pacific Northwest. *Conservation Biology* 3:389-396.

Doerr, P. D., J. R. Walters, and J. H. Carter, III. 1989. Reoccupation of abandoned clusters of cavity trees (colonies) by red-cockaded woodpeckers. *Proceedings of the Annual Conference of the Southeastern Association of Fish and Wildlife Agencies* 43:326-336.

Doherty, P. F., Jr., E. A. Marschall, and T. C. Grubb, Jr. 1999. Balancing conservation and economic gain: a dynamic programming approach. *Ecological Economics* 29:349-358.

Donovan, T. M., F. R. Thompson, III, J. Faaborg, and J. R. Probst. 1995. Reproductive success of migratory birds in habitat sources and sinks. *Conservation Biology* 9:1380-1395.

Doster, R. H., and D. A. James. 1998. Home range size and foraging habitat of Red-cockaded Woodpeckers in the Ouachita Mountains of Arkansas. *Wilson Bulletin* 110:110-117.

Dreyfus, S. E., and A. M. Law. 1977. *The art and theory of dynamic programming*. Academic Press, New York, New York, USA.

Duguay, J. P., P. B. Wood, and J. V. Nichols. 2001. Songbird abundance and avian nest survival rates in forests fragmented by different silvicultural treatments. *Conservation Biology* 15:1405-1415.

Dunning, J. B., Jr., D. J. Stewart, B. J. Danielson, B. R. Noon, T. L. Root, R. H. Lamberson, and E. E. Stevens. 1995. Spatially explicit population models: current forms and future uses. *Ecological Applications* 5:3-11.

Fauth, P. T., E. J. Gustafson, and K. N. Rabenold. 2000. Using landscape metrics to model source habitat for Neotropical migrants in the midwestern U.S. *Landscape Ecology* 15:621-631.

Flather, C. H., and R. M. King. 1992. Evaluating performance of regional wildlife habitat models: implications to resource planning. *Journal of Environmental Management* 34:31-46.

Forsman, E. D., E. C. Meslow, and H. M. Wight. 1984. Distribution and biology of the spotted owl in Oregon. *Wildlife Monographs* 87:1-64.

Friesen, L. E., M. D. Cadman, and R. J. MacKay. 1999. Nesting success of Neotropical migrant songbirds in a highly fragmented landscape. *Conservation Biology* 13:338-346.

Frost, C. C. 1998. Presettlement fire frequency regimes of the United States: a first approximation. *Tall Timbers Fire Ecology Conference Proceedings* 20:70-81.

Gabrielson, I. N. 1943. *Wildlife refuges*. Macmillan, New York, New York, USA.

Goldberg, D. E. 1989. *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley, Reading, Massachusetts, USA.

Grumbine, R. E. 1994. What is ecosystem management? *Conservation Biology* 8:27-38.

Haight, R. G. 1995. Comparing extinction risk and economic cost in wildlife conservation planning. *Ecological Applications* 5:767-775.

Haight, R. G., and L. E. Travis. 1997. Wildlife conservation planning using stochastic optimization and importance sampling. *Forest Science* 43:129-139.

Hamel, P. B., H. E. LeGrand, Jr., M. R. Lennartz, and S. A. Gauthreaux, Jr. 1982. Bird-habitat relationships on Southeastern forest lands. USDA Forest Service, Southeastern Forest Experiment Station, General Technical Report SE-22.

Hansen, A. J., S. L. Garman, J. F. Weigand, D. L. Urban, W. C. McComb, and M. G. Raphael. 1995. Alternative silvicultural regimes in the Pacific Northwest: simulations of ecological and economic effects. *Ecological Applications* 5:535-554.

Hansen, A. J., and D. L. Urban. 1992. Avian response to landscape pattern: the role of species' life histories. *Landscape Ecology* 7:163-180.

Hepinstall, J. A., and S. A. Sader. 1997. Using Bayesian statistics, Thematic Mapper satellite imagery, and Breeding Bird Survey data to model bird species probability of occurrence in Maine. *Photogrammetric Engineering and Remote Sensing* 63:1231-1237.

Heppell, S. S., J. R. Walters, and L. B. Crowder. 1994. Evaluating management alternatives for red-cockaded woodpeckers: a modeling approach. *Journal of Wildlife Management* 58:479-487.

Hess, C. A., and F. C. James. 1998. Diet of the red-cockaded woodpecker in the Apalachicola National Forest. *Journal of Wildlife Management* 62:509-517.

Hines, M. 1999. Managing red-cockaded woodpeckers (*Picoides borealis*) affects breeding bird communities of pine-oak forests in southeastern Kentucky. *Journal of the Kentucky Academy of Science* 60:78-86.

Hof, J., M. Bevers, L. Joyce, and B. Kent. 1994. An integer programming approach for spatially and temporally optimizing wildlife populations. *Forest Science* 40:177-191.

Hof, J., and M. G. Raphael. 1997. Optimization of habitat placement: a case study of the Northern Spotted Owl in the Olympic Peninsula. *Ecological Applications* 7:1160-1169.

Holling, C. S., and G. K. Meffe. 1996. Command and control and the pathology of natural resource management. *Conservation Biology* 10:328-337.

Holt, R. D., S. W. Pacala, T. W. Smith, and J. Liu. 1995. Linking contemporary vegetation models with spatially explicit animal population models. *Ecological Applications* 5:20-27.

Hooper, R. G., and C. J. McAdie. 1995. Hurricanes and the long-term management of the red-cockaded woodpecker. Pages 148-166 *in* D. L. Kulhavy, R. G. Hooper, and R. Costa, editors. *Red-cockaded woodpecker: recovery, ecology and management*. Center for Applied Studies in Forestry, Stephen F. Austin State University, Nacogdoches, Texas, USA.

Hoover, J. P., and M. C. Brittingham. 1998. Nest-site selection and nesting success of Wood Thrushes. *Wilson Bulletin* 110:375-383.

Hovis, J. A., and R. F. Labisky. 1985. Vegetative associations of red-cockaded woodpecker colonies in Florida. *Wildlife Society Bulletin* 13:307-314.

Hughell, D. A. 1996. Simulated adaptive management for timber and wildlife under uncertainty. Dissertation, North Carolina State University, Raleigh, North Carolina, USA.

Hunter, W. C., D. N. Pashley, and R. E. F. Escano. 1992. Neotropical migratory landbird species and their habitats of special concern within the Southeast Region. Pages 159-171 *in* D. M. Finch and P. W. Stangel, editors. *Status and management of neotropical migratory birds*. USDA Forest Service General Technical Report RM-229.

Interagency Ecosystem Management Task Force. 1995. The ecosystem approach: healthy ecosystems and sustainable economies. National Technical Information Service, Springfield, Virginia, USA.

Irwin, L. L., and T. B. Wigley. 1993. Toward an experimental basis for protecting forest wildlife. *Ecological Applications* 3:213-217.

Jackson, J. A. 1994. Red-cockaded woodpecker (*Picoides borealis*). Pages 1-20 *in* A. Poole and F. Gill, editors. *Birds of North America*. No. 85. Academy of Natural Sciences, Philadelphia, Pennsylvania and American Ornithologists' Union, Washington, D.C., USA.

James, F. C., C. A. Hess, B. C. Kicklighter, and R. A. Thum. 2001. Ecosystem management and the niche gestalt of the Red-cockaded Woodpecker in longleaf pine forests. *Ecological Applications* 11:854-870.

James, F. C., C. A. Hess, and D. Kufrin. 1997. Species-centered Environmental Analysis: indirect effects of fire history on Red-cockaded Woodpeckers. *Ecological Applications* 7:118-129.

James, F. C., R. F. Johnston, N. O. Wamer, G. J. Niemi, and W. J. Boecklen. 1984. The Grinnellian niche of the wood thrush. *American Naturalist* 124:17-47.

James, F. C. and H. H. Shugart. 1970. A quantitative method of habitat description. *Audubon Field Notes* 24:727-736.

Jensen, M. E., P. Bourgeron, R. Everett, and I. Goodman. 1996. Ecosystem management: a landscape ecology perspective. *Water Resources Bulletin* 32:203-216.

Johnson, F. A., C. T. Moore, W. L. Kendall, J. A. Dubovsky, D. F. Caithamer, J. R. Kelley, Jr., and B. K. Williams. 1997. Uncertainty and the management of mallard harvests. *Journal of Wildlife Management* 61:202-216.

Johnson, F. A., and B. K. Williams. 1999. Protocol and practice in the adaptive management of waterfowl harvests. *Conservation Ecology* 3(1):8. [online] URL: http://www.consecol.org/vol3/iss1/art8

Johnson, F. A., B. K. Williams, J. D. Nichols, J. E. Hines, W. L. Kendall, G. W. Smith, and D. F. Caithamer. 1993. Developing an adaptive management strategy for harvesting waterfowl in North America. *Transactions of the North American Wildlife and Natural Resources Conference* 58:565-583.

Johnston, D. W., and E. P. Odum. 1956. Breeding bird populations in relation to plant succession on the Piedmont of Georgia. *Ecology* 37:50-62.

Jones, E. P., Jr. 1993. Silvicultural treatments to maintain red-cockaded woodpecker habitat. Pages 627-632 *in* J. C. Brissette, ed. *Proceedings of the seventh biennial southern silvicultural research conference*. USDA Forest Service General Technical Report SO-93.

Kalisz, P. J., and S. E. Boettcher. 1991. Active and abandoned red-cockaded woodpecker habitat in Kentucky. *Journal of Wildlife Management* 55:146-154.

Keiter, R. B. 1998. Ecosystems and the law: toward an integrated approach. *Ecological Applications* 8:332-341.

Keller, C. M. E., C. S. Robbins, and J. S. Hatfield. 1993. Avian communities in riparian forests of different widths in Maryland and Delaware. *Wetlands* 13:137-144.

Kurttila, M. 2001. The spatial structure of forests in the optimization calculations of forest planning – a landscape ecological perspective. *Forest Ecology and Management* 142:129-142.

Lancia, R. A., C. E. Braun, M. W. Collopy, R. D. Dueser, J. G. Kie, C. J. Martinka, J. D. Nichols, T. D. Nudds, W. R. Porath, and N. G. Tilghman. 1996. ARM! For the future: adaptive resource management in the wildlife profession. *Wildlife Society Bulletin* 24:436-442.

Lennartz, M. R., and D. G. Heckel. 1987. Population dynamics of a red-cockaded woodpecker population in Georgia Piedmont loblolly pine habitat. Pages 48-55 *in* R. R. Odom, K. A. Riddleberger, and J. C. Ozier, editors. *Proceedings of the third southeastern nongame and endangered wildlife symposium*. Georgia Department of Natural Resources, Social Circle, Georgia, USA.

Letcher, B. H., J. A. Priddy, J. R. Walters, and L. B. Crowder. 1998. An individual-based, spatially-explicit simulation model of the population dynamics of the endangered red-cockaded woodpecker, *Picoides borealis*. *Biological Conservation* 86:1-14.

Ligon, J. D., P. B. Stacey, R. N. Conner, C. E. Bock, and C. S. Adkisson. 1986. Report of the American Ornithologists' Union Committee for the Conservation of the Red-cockaded Woodpecker. *Auk* 103:848-855.

Lin, C. R., and J. Buongiorno. 1998. Tree diversity, landscape diversity, and economics of maple-birch forests: implications of Markovian models. *Management Science* 44:1351-1366.

Lindley, D. V. 1985. *Making decisions*. Second edition. John Wiley and Sons, London, UK.

Loeb, S. C., W. D. Pepper, and A. T. Doyle. 1992. Habitat characteristics of active and abandoned red-cockaded woodpecker colonies. *Southern Journal of Applied Forestry* 16:120-125.

Loehle, C. 2000. Optimal control of spatially distributed process models. *Ecological Modelling* 131:79-95.

Long, D. D., and M. E. Carr. 1916. Soil survey of Jasper County, Georgia. USDA Bureau of Soils, Washington, D.C., USA.

Lubow, B. C. 1995. SDP: generalized software for solving stochastic dynamic optimization problems. *Wildlife Society Bulletin* 23:738-742.

Lubow, B. C. 1996. Optimal translocation strategies for enhancing stochastic metapopulation viability. *Ecological Applications* 6:1268-1280.

Lucas, K. E. 1994. Modelling avian responses to red-cockaded woodpecker habitat management in loblolly pine forests of east-central Mississippi. Thesis, Mississippi State University, Mississippi State, Mississippi, USA.

Lynch, J. F., and D. F. Whigham. 1984. Effects of forest fragmentation on breeding bird communities in Maryland, USA. *Biological Conservation* 28:287-324.

Maguire, L. A., G. F. Wilhere, and Q. Dong. 1995. Population viability analysis for red-cockaded woodpeckers in the Georgia Piedmont. *Journal of Wildlife Management* 59:533-542.

Malone, C. R. 2000. Ecosystem management policies in state government of the USA. *Landscape and Urban Planning* 48:57-64.

Mancke, R. G., and T. A. Gavin. 2000. Breeding bird density in woodlots: effects of depth and buildings at the edges. *Ecological Applications* 10:598-611.

Marzluff, J. M., M. G. Raphael, and R. Sallabanks. 2000. Understanding the effects of forest management on avian species. *Wildlife Society Bulletin* 28:1132-1143.

Masters, R. E., C. W. Wilson, G. A. Bukenhofer, and M. E. Payton. 1996. Effects of pine-grassland restoration for red-cockaded woodpeckers on white-tailed deer forage production. *Wildlife Society Bulletin* 24:77-84.

McAllister, M. K., and G. P. Kirkwood. 1998. Bayesian stock assessment: a review and example application using the logistic model. *ICES Journal of Marine Science* 55:1031-1060.

Meyers, J. M., and A. S. Johnson. 1978. Bird communities associated with succession and management of loblolly-shortleaf pine forests. Pages 60-65 *in* R. M. DeGraaf, technical coordinator. *Proceedings of the workshop management of southern forests for nongame birds*. USDA Forest Service General Technical Report SE-14.

Mitchell, M. S., R. A. Lancia, and J. A. Gerwin. 2001. Using landscape-level data to predict the distribution of birds on a managed forest: effects of scale. *Ecological Applications* 11:1692-1708.

Moore, C. T., M. J. Conroy, and K. Boston. 2000. Forest management decisions for wildlife objectives: system resolution and optimality. *Computers and Electronics in Agriculture* 27:25-39.

Mowrer, H. T. 2000. Uncertainty in natural resource decision support systems: sources, interpretation, and importance. *Computers and Electronics in Agriculture* 27:139-154.

Murphy, D. D., and B. R. Noon. 1992. Integrating scientific methods with habitat conservation planning: reserve design for Northern Spotted Owls. *Ecological Applications* 2:3-17.

Myers, N. 1993. Biodiversity and the precautionary principle. *Ambio* 22:74-79.

Nelson, G., P. Lawrence, and H. Black. 2000. Assessing ecosystem conservation plans for Canadian national parks. *Natural Areas Journal* 20:280-287.

Nelson, T. C. 1957. The original forests of the Georgia Piedmont. *Ecology* 38:390-397.

Nevo, A., and L. Garcia. 1996. Spatial optimization of wildlife habitat – model formulation. *Transactions of the American Society of Agricultural Engineers* 39:1189-1196.

Nichols, J. D., M. J. Conroy, D. R. Anderson, and K. P. Burnham. 1984. Compensatory mortality in waterfowl populations: a review of the evidence and implications for research and management. *Transactions of the North American Wildlife and Natural Resources Conference* 49:535-554.

Nichols, J. D., F. A. Johnson, and B. K. Williams. 1995. Managing North American waterfowl in the face of uncertainty. *Annual Review of Ecology and Systematics* 26:177-199.

Noon, B. R. 1981. Techniques for sampling avian habitats. Pages 42-52 *in* D. E. Capen, editor. *Proceedings of the use of multivariate statistics in studies of wildlife habitat: a workshop*. School of Natural Resources, University of Vermont, U. S. Fish and Wildlife Service, and USDA Forest Service. Burlington, Vermont, USA.

Noon, B. R., and J. R. Sauer. 1992. Population models for passerine birds: structure, parameterization, and analysis. Pages 441-464 *in* D. R. McCullough and R. H. Barrett, editors. *Wildlife 2001: populations*. Elsevier Applied Science, London, UK.

Pascual, M. A., and R. Hilborn. 1995. Conservation of harvest populations in fluctuating environments: the case of the Serengeti wildebeest. *Journal of Applied Ecology* 32:468-480.

Pascual, M. A., P. Kareiva, and R. Hilborn. 1997. The influence of model structure on conclusions about the viability and harvesting of Serengeti wildebeest. *Conservation Biology* 11:966-976.

Payne, H. H. 1976. Soil survey of Baldwin, Jones, and Putnam Counties, Georgia. USDA Soil Conservation Service and USDA Forest Service, Washington, D.C., USA.

Penhollow, M. E., and D. F. Stauffer. 2000. Large-scale habitat relationships of Neotropical migratory birds in Virginia. *Journal of Wildlife Management* 64:362-373.

Peterjohn, B. G., J. R. Sauer, and C. S. Robbins. 1995. Population trends from the North American Breeding Bird Survey. Pages 3-39 *in* T. E. Martin and D. M. Finch, editors. *Ecology and management of Neotropical migratory birds*. Oxford University Press, New York, New York, USA.

Peterman, R. M., and J. L. Anderson. 1999. Decision analysis: a method for taking uncertainties into account in risk-based decision making. *Human and Ecological Risk Assessment* 5:231-244.

Peterson, C. J. 2000. Catastrophic wind damage to North American forests and the potential impact of climate change. *Science of the Total Environment* 262:287-311.

Powell, L. A. 1998. Experimental analysis and simulation modeling of forest management impacts on wood thrushes, Hylocichla mustelina. Dissertation, University of Georgia, Athens, Georgia, USA.

Powell, L. A., J. D. Lang, M. J. Conroy, and D. G. Krementz. 2000. Effects of forest management on density, survival, and population growth of wood thrushes. *Journal of Wildlife Management* 64:11-23.

Pulliam, H. R. 1988. Sources, sinks, and population regulation. *The American Naturalist* 132:652-661.

Pulliam, H. R., J. B. Dunning, Jr., and J. Liu. 1992. Population dynamics in complex landscapes: a case study. *Ecological Applications* 2:165-177.

Radomski, P. J., and T. J. Goeman. 1996. Decision making and modeling in freshwater sport-fisheries management. *Fisheries* 21(12):14-21.

Raftovich, R. V., Jr. 1998. Effects of management for Red-cockaded Woodpeckers on avian communities in mature pine stands in the Georgia Piedmont. Thesis, University of Georgia, Athens, Georgia, USA.

Rauscher, H. M. 1999. Ecosystem management decision support for federal forests in the United States: a review. *Forest Ecology and Management* 114:173-197.

Reed, J. M., J. R. Walters, T. E. Emigh, and D. E. Seaman. 1993. Effective population size in Red-cockaded Woodpeckers: population and model differences. *Conservation Biology* 7:302-308.

Rieman, B., J. T. Peterson, J. Clayton, P. Howell, R. Thurow, W. Thompson, and D. Lee. 2001. Evaluation of potential effects of federal land management alternatives on trends of salmonids and their habitats in the interior Columbia River basin. *Forest Ecology and Management* 153:43-62.

Robbins, C. S., D. K. Dawson, and B. A. Dowell. 1989. Habitat area requirements of breeding forest birds of the Middle Atlantic States. *Wildlife Monographs* 103:1-34.

Rodewald, P. G., and K. G. Smith. 1998. Short-term effects of understory and overstory management on breeding birds in Arkansas oak-hickory forests. *Journal of Wildlife Management* 62:1411-1417.

Rohweder, M. R., C. W. McKetta, and R. A. Riggs. 2000. Economic and biological compatibility of timber and wildlife production: an illustrative use of production possibilities frontier. *Wildlife Society Bulletin* 28:435-447.

Roth, R. R., M. S. Johnson, and T. J. Underwood. 1996. Wood Thrush (*Hylocichla mustelina*). Pages 1-28 *in* A. Poole and F. Gill, editors. *Birds of North America*. No. 246. Academy of Natural Sciences, Philadelphia, Pennsylvania and American Ornithologists' Union, Washington, D.C., USA.

Ruckelshaus, M., C. Hartway, and P. Kareiva. 1997. Assessing the data requirements of spatially explicit dispersal models. *Conservation Biology* 11:1298-1306.

Schweder, T. 2001. Protecting whales by distorting uncertainty: non-precautionary mismanagement? *Fisheries Research* 52:217-225.

Seagle, S. W., R. A. Lancia, D. A. Adams, M. R. Lennartz, and H. A. Devine. 1987. Integrating timber and red-cockaded woodpecker habitat management.

Transactions of the North American Wildlife and Natural Resources Conference 52:41-52.

Shaffer, M. L. 1981. Minimum population sizes for species conservation. *BioScience* 31:131-134.

Sherry, T. W., and R. T. Holmes. 1992. Are populations of Neotropical migrant birds limited in summer or winter? Implications for management. Pages 47-57 *in* D. M. Finch and P. W. Stangel, editors. *Status and management of neotropical migratory birds*. USDA Forest Service General Technical Report RM-229.

Sherry, T. W., and R. T. Holmes. 1995. Summer versus winter limitation of populations: what are the issues and what is the evidence? Pages 85-120 *in* T. E. Martin and D. M. Finch, editors. *Ecology and management of Neotropical migratory birds*. Oxford University Press, New York, New York, USA.

Simons, T. R., G. L. Farnsworth, and S. A. Shriner. 2000. Evaluating Great Smoky Mountains National Park as a population source for the Wood Thrush. *Conservation Biology* 14:1133-1144.

Smith, D. M. 1962. *The practice of silviculture*. Seventh edition. John Wiley and Sons, New York, New York, USA.

Stage, A. R. 1976. An expression for the effect of aspect, slope, and habitat type on tree growth. *Forest Science* 22:457-460.

Stevens, E. E. 1995. Population viability considerations for red-cockaded woodpecker recovery. Pages 227-238 *in* D. L. Kulhavy, R. G. Hooper, and R. Costa, editors. *Red-cockaded woodpecker: recovery, ecology and management*. Center for Applied Studies in Forestry, Stephen F. Austin State University, Nacogdoches, Texas, USA.

Temple, S. A., and J. R. Cary. 1988. Modeling dynamics of habitat-interior bird populations in fragmented landscapes. *Conservation Biology* 2:340-347.

Thomas, J. W., E. D. Forsman, J. B. Lint, E. C. Meslow, B. R. Noon, and J. Verner. 1990. A conservation strategy for the northern spotted owl. USDA Forest Service, Bureau of Land Management, U.S. Fish and Wildlife Service, and National Park Service, Portland, Oregon, USA.

Thomas, L., J. L. Laake, J. F. Derry, S. T. Buckland, D. L. Borchers, D. R. Anderson, K. P. Burnham, S. Strindberg, S. L. Hedley, M. L. Burt, F. F. C. Marques, J. H. Pollard, and R. M. Fewster. 1998. *Distance 3.5*. Research Unit for Wildlife Population Assessment, University of St. Andrews, St. Andrews, UK. [online] URL: http://www.ruwpa.st-and.ac.uk/distance/

Thomlinson, J. R. 1995. Landscape characteristics associated with active and abandoned Red-cockaded Woodpecker clusters in East Texas. *Wilson Bulletin* 107:603-614.

Thomlinson, J. R. 1996. Predicting status change in red-cockaded woodpecker cavity-tree clusters. *Journal of Wildlife Management* 60:350-354.

Thompson, F. R., III, W. D. Dijak, T. G. Kulowiec, and D. A. Hamilton. 1992. Breeding bird populations in Missouri Ozark forests with and without clearcutting. *Journal of Wildlife Management* 56:23-30.

Thompson, F. R., III, J. R. Probst, and M. G. Raphael. 1995. Impacts of silviculture: overview and management recommendations. Pages 201-219 *in* T. E. Martin and D. M. Finch, editors. *Ecology and management of Neotropical migratory birds*. Oxford University Press, New York, New York, USA.

Tuck, G. N., and H. P. Possingham. 2000. Marine protected areas for spatially structured exploited stocks. *Marine Ecology Progress Series* 192:89-101.

U.S. Fish and Wildlife Service. 1976. Final environmental statement: operation of the National Wildlife Refuge System. U.S. Department of Interior, Fish and Wildlife Service, Washington, D.C., USA.

U.S. Fish and Wildlife Service. 1985. Red-cockaded woodpecker recovery plan. U.S. Department of Interior, Fish and Wildlife Service, Atlanta, Georgia, USA.

U.S. Fish and Wildlife Service. 1988. Draft environmental impact statement: management of the national wildlife refuges. U.S. Department of Interior, Fish and Wildlife Service, Washington, D.C., USA.

U.S. Fish and Wildlife Service. 1995. Migratory nongame birds of management concern in the United States: the 1995 list. U.S. Department of Interior, Fish and Wildlife Service, Office of Migratory Bird Management, Washington, D.C., USA

U.S. Fish and Wildlife Service. 1998. Strategy and guidelines for the recovery and management of the red-cockaded woodpecker and its habitats on national wildlife refuges. U.S. Department of Interior, Fish and Wildlife Service, Atlanta, Georgia, USA.

U.S. Fish and Wildlife Service. 2000. Technical/agency draft revised recovery plan for the red-cockaded woodpecker (Picoides borealis). U.S. Department of Interior, Fish and Wildlife Service, Atlanta, Georgia, USA.

Valverde, T., and J. Silvertown. 1997. Canopy closure rate and forest structure. *Ecology* 78:1555-1562.

Van Balen, J. B., and P. D. Doerr. 1978. The relationship of understory vegetation to red-cockaded woodpecker activity. *Proceedings of the Annual Conference of the Southeastern Association of Fish and Wildlife Agencies* 32:82-92.

Van Horne, B. 1983. Density as a misleading indicator of habitat quality. *Journal of Wildlife Management* 47:893-901.

Verner, J., M. L. Morrison, and C. J. Ralph, editors. 1986. *Wildlife 2000: modeling habitat relationships of terrestrial vertebrates*. University of Wisconsin Press, Madison, Wisconsin, USA.

Walters, C. J. 1986. *Adaptive management of renewable resources*. Macmillan, New York, New York, USA.

Walters, C. J., and R. Hilborn. 1978. Ecological optimization and adaptive management. *Annual Review of Ecology and Systematics* 9:157-188.

Walters, C. J., and C. S. Holling. 1990. Large-scale management experiments and learning by doing. *Ecology* 71:2060-2068.

Walters, J. R. 1991. Application of ecological principles to the management of endangered species: the case of the red-cockaded woodpecker. *Annual Review of Ecology and Systematics* 22:505-523.

Walters, J. R., L. B. Crowder, and J. A. Priddy. 2002. Population viability analysis for Red-cockaded Woodpeckers using an individual-based model. *Ecological Applications* 12:249-260.

Weaver, F. G. 1949. Hylocichla mustelina: wood thrush. Pages 101-123 *in* A. C. Bent, editor. *Life histories of North American thrushes, kinglets, and their allies*. U.S. National Museum Bulletin 196. [republished 1964 by Dover, New York, New York, USA]

Whitcomb, R. F., C. S. Robbins, J. F. Lynch, B. L. Whitcomb, M. K. Klimkiewicz, and D. Bystrak. 1981. Effects of forest fragmentation on avifauna of the eastern deciduous forest. Pages 125-205 *in* R. L. Burgess and D. M. Sharpe, editors. *Forest island dynamics in man-dominated landscapes*. Springer-Verlag, New York, New York, USA.

Wigley, T. B., S. W. Sweeney, and J. R. Sweeney. 1999. Habitat attributes and reproduction of red-cockaded woodpeckers in intensively managed forests. *Wildlife Society Bulletin* 27:801-809.

Williams, B. K. 1982. Optimal stochastic control in natural resource management: framework and examples. *Ecological Modelling* 16:275-297.

Williams, B. K. 1989. Review of dynamic optimization methods in renewable natural resource management. *Natural Resource Modeling* 3:137-216.

Williams, B. K. 1996. Adaptive optimization and the harvest of biological populations. *Mathematical Biosciences* 136:1-20.

Williams, B. K. 1997. Approaches to the management of waterfowl under uncertainty. *Wildlife Society Bulletin* 25:714-720.

Williams, B. K. 2001. Uncertainty, learning, and the optimal management of wildlife. *Environmental and Ecological Statistics* 8:269-288.

Williams, B. K., F. A. Johnson, and K. Wilkins. 1996. Uncertainty and the adaptive management of waterfowl harvests. *Journal of Wildlife Management* 60:223-232.

Williams, B. K., and J. D. Nichols. 2001. Systems identification and the adaptive management of waterfowl in the United States. *Wildlife Biology* 7:223-236.

Wilson, C. W., R. E. Masters, and G. A. Bukenhofer. 1995. Breeding bird response to pine-grassland community restoration for red-cockaded woodpeckers. *Journal of Wildlife Management* 59:56-57.

APPENDIX A

TABLES

Table 1. Parameters and parameter values used in Refuge management simulation model.

Parameter	Description	Value
Basal area growth model and modifications (Bailey and Ware 1983)		
THININDX	Thinning type	0.6
B1	Age coefficient	2.81706
B2	Thinning coefficient	-11935.2
B3	Site index coefficient	0.043493
AGE0	Model discontinuity point	8
BA0	Basal area for age<AGE0	0.46
BA_MAX	Basal area max	35
Canopy closure model (Valverde and Silvertown 1997)		
CCRATE	Annual rate of canopy closure	0.79
Understory vegetation density model and modifications (Conroy et al. 1982)		
V1	Intercept	-1461.4
V2	Basal area coefficient	36554.6
V3	Canopy cover coefficient	21.4
V4	Slope position coefficient	100.0
V5	Canopy cover-slope position interaction	-14.8
V6	Basal area-slope position interaction	13.8
BURNPARM1	Vegetation reduction 1 year post-burn	0.76
BURNPARM2	Vegetation reduction 2 years post-burn	0.87
BURNPARM3	Vegetation reduction 3 years post-burn	0.97
Regeneration patch selection parameters		
WT_AREA	Weight for area component of selection criterion	1
WT_DIST	Weight for distance component of selection criterion	1
CUTLIMIT	Patch size limit (ha)	10.1171
Thinning selection parameters		
BA_THIN	Residual basal area (m^2/ha)	13.7741
BA_CEIL	Ceiling basal area (m^2/ha)	18.3655
THIN_AGE	Min age eligible for thinning	16
800-m radius foraging habitat parameters		
FH1_AGE	Min stand age	40
FH1_BA1	Min basal area (m^2/ha)	9.1827
FH1_BA2	Max basal area (m^2/ha)	18.3655
FH1_AREA	Min area of habitat in buffer (ha)	50.5857
OUT800_1	Portion of adjacent ONF site providing foraging habitat	0.5
OUT800_2	Portion of adjacent private site providing foraging habitat	0.2

Table 1. Continued.

Parameter	Description	Value
Recruitment cluster placement		
CLUSGROW	Target growth rate of active clusters	0.10
STRMDIST	Min distance (m) of cluster to stream	125
RC_AGE1	Nesting habitat min age requirement 1	40
RC_AMT1	Min proportion of cell RC_AGE1 or older	0.85
RC_AGE2	Nesting habitat min age requirement 2	60
RC_AMT2	Min proportion of cell RC_AGE2 or older	0.50
CLUS_D1	Min nearest-neighbor distance (m) to any cluster	402.336
CLUS_D2A	Max desired distance (m) from any active cluster	1609.344
CLUS_D2B	Max permissible distance (m) from any active cluster	3218.688
CLUS_INC	Relaxation increment (m) for distance test	402.336
Hardwood encroachment parameters		
BURNLIM	Period (yrs) of no encroachment following burning	5
THINLIM	Period (yrs) of no encroachment following thinning	5
HWD_P1	Annual avg rate of P1 pine loss to hardwood	0.0006
HWD_P2	Annual avg rate of P2 pine loss to hardwood	0.015
HWD_P3	Annual avg rate of P3 pine loss to hardwood	0.015
HWD_P4	Annual avg rate of P4 pine loss to hardwood	0.03
HWD_CV	CV (%) for random encroachment rate	40
Forest disturbance parameters		
DSTRB_MN	Mean rate (area proportion) of disturbance	0.003394
DSTRB_VR	Variance of disturbance proportion	0.0003572
DSTRB_HW	Portion of upland hardwood regenerating as pine	0.25
Woodpecker cluster occupancy model and modifications (Loeb et al. 1992)		
LOEBPARM1	Intercept for occupancy model	5.134
LOEBPARM2	Slope for occupancy model	-0.45738
BA_ADJ1	Scale parameter for persistence adjustment	1.2
BA_ADJ2	Constant for persistence adjustment	8.26446
TBA	Exponent for vegetation density conversion	0.66667
AVG_TVD	Mean transformed vegetation density (kg/ha) in old stands	85.132063
AVG_MBA	Weighted mean midstory basal area (m^2/ha)	9.174073

Table 1. Continued.

Parameter	Description	Value
Red-cockaded woodpecker productivity parameters		
FLEDG_DIST	Distribution of fledgling group size	(Table 2)
FA_0	Intercept value for ψ function (ha)	75
SURV_FL	Fledgling survival to adulthood	0.380
ZETA	ζ value for alternative habitat-sensitive model	0.8
ALPHA	α value for alternative habitat-sensitive model	2
A	Scale parameter for recruitment isolation effect	-0.69077
R0	Constant for recruitment isolation effect	0.080979
HEF_X	X-coordinate (UTM) of approximate HEF center	250500
HEF_Y	Y-coordinate (UTM) of approximate HEF center	3661000
HEF_PROD	Annual recruitment size of HEF clusters	6
Wood thrush regression model parameters (variances and covariances not reported)		
DB12A_1	Intercept for DB12A model	0.34823
DB12A_2	Time-since-burn slope for DB12A model	0.02332
DB12L_1	Intercept for DB12L model	-0.59658
DB12L_2	Time-since-burn slope for DB12L model	0.03339
DB12L_3	Vegetation density slope for DB12L model	0.00015544
DH_1	Intercept for D_H model	18.89173
DH_2	Basal area slope for D_H model	-0.45437
DH_3	Canopy closure slope for D_H model	-26.77392
DH_4	DB12A slope for D_H model	-1.82827
DH_5	Basal area-canopy closure interaction for D_H model	0.55391
DP_1	Intercept for D_P model	-7.80291
DP_2	Basal area slope for D_P model	0.09640
DP_3	DB12A slope for D_P model	-4.65806
DP_4	DB12L slope for D_P model	-14.80672
DP_5	Basal area-DB12L interaction for D_P model	0.25183
DP_6	DB12A-DB12L interaction for D_P model	-5.99684

Table 2. Calculations of group size distributions per active cluster for nestlings ($p_n(x)$), fledglings within nestling size class ($p_{f|n}(y \mid x)$), fledglings unconditional on nestling size class ($p_f(x)$), and recruits into adulthood ($p_r(x)$) of red-cockaded woodpeckers on the Piedmont National Wildlife Refuge, 1990-2000.

Size class	Nestlings (1990-2000)		Fledglings by nestling size classes (1990-1995)												Size class distributions	
			Counts in fledgling size class y						Conditional frequencies $p_{f\|n}(y \mid x)$						
x	n	$p_n(x)$	0	1	2	3	4	Total	0	1	2	3	4	$p_f(x)$	$p_r(x)$
0	110	0.336	40					40	1					0.405	0.6272
1	23	0.070	0	13				13	0	1				0.115	0.2607
2	73	0.223	3	3	23			29	0.103	0.103	0.793			0.264	0.0977
3	90	0.275	5	3	11	26		45	0.111	0.067	0.244	0.578		0.192	0.0139
4	28	0.086	3	0	4	8	6	21	0.143	0	0.190	0.381	0.286	0.024	0.0005
5	2	0.006	1	0	1	0	0	2	0.500	0	0.500	0	0	0	0
6	1	0.003	0	1	0	0	0	1	0	1	0	0	0	0	0

Table 3. Estimated mean ($\bar{x}$) and standard error (SE) of wood thrush density (birds/ha), by year and overstory cover type, in surveyed compartments of the Piedmont National Wildlife Refuge, 1998-1999.

	Year							
	1998				1999			
	Overstory Cover Type				Overstory Cover Type			
	Hardwood		Pine		Hardwood		Pine	
Compartment	$\bar{x}$	SE	$\bar{x}$	SE	$\bar{x}$	SE	$\bar{x}$	SE
12	0.0204	0.0065	0.0102	0.0046	0.0561	0.0120	0.0357	0.0095
16	0.0067	0.0039	0.0357	0.0089	0.0479	0.0157	0.0360	0.0132
23	0.0115	0.0056	0.0390	0.0120	0.0397	0.0099	0.0298	0.0086
24	0.0105	0.0053	0.0368	0.0098	0.0381	0.0114	0.0871	0.0193
25	0.0307	0.0074	0.0108	0.0044	0.0929	0.0235	0.0221	0.0103
27	0.0158	0.0064	0.0158	0.0064	0.0298	0.0105	0.0149	0.0074
31	0.0339	0.0102	0.0308	0.0097	0.1084	0.0231	0.0542	0.0163
33	0.0197	0.0070	0.0222	0.0074	0.0197	0.0070	0.0222	0.0074

Table 4. Elasticity values for forest overstory model, estimated for five model responses (proportions of forest in five cover types) at each of three alternative settings (0.5×, 1.0×, 2.0×) of each model parameter. Values of elasticity exceeding 0.20 in absolute value appear in boldface for emphasis.

	Model responses[a]														
	P1			P2			P3			P4			UH		
Parameter	0.5×	1.0×	2.0×	0.5×	1.0×	2.0×	0.5×	1.0×	2.0×	0.5×	1.0×	2.0×	0.5×	1.0×	2.0×
τ_{21}	**-0.36**	**-0.45**	**-0.56**	**0.63**	**0.54**	**0.44**	**0.62**	**0.54**	**0.44**	**0.62**	**0.54**	**0.43**	0.08	0.04	-0.02
τ_{32}	-0.14	-0.16	-0.12	**-0.29**	**-0.40**	**-0.55**	**0.71**	**0.60**	**0.45**	**0.70**	**0.60**	**0.45**	0.12	0.11	0.09
τ_{43}	0.02	0.03	-0.02	0.02	0.04	-0.01	**-0.31**	**-0.44**	**-0.59**	**0.69**	**0.56**	**0.41**	0.07	0.05	0.08
τ_{14}	0.03	0.04	0.06	0.03	0.04	0.06	0.03	0.04	0.06	**-0.22**	**-0.35**	**-0.53**	-0.03	-0.05	-0.07
$\mu(e_1)$	-0.05	-0.11	**-0.23**	-0.05	-0.11	**-0.23**	-0.05	-0.11	**-0.23**	-0.05	-0.10	**-0.22**	0.10	0.19	**0.27**
$\mu(e_2)$	0.00	-0.10	**-0.41**	0.00	-0.11	**-0.59**	0.00	-0.11	**-0.51**	0.00	-0.11	**-0.47**	0.00	0.18	**0.48**
$\mu(e_3)$	-0.04	**-0.36**	**-0.91**	-0.04	**-0.36**	**-0.91**	-0.04	**-0.43**	**-1.02**	-0.05	**-0.43**	**-1.04**	0.09	**0.62**	**0.67**
$\mu(e_4)$	-0.03	**-0.22**	**-0.34**	-0.03	**-0.21**	**-0.34**	-0.03	**-0.20**	**-0.33**	-0.06	**-0.54**	**-1.00**	0.07	**0.37**	**0.30**
v_e	-0.08	**-0.30**	**-1.01**	-0.08	**-0.30**	**-1.11**	-0.09	**-0.34**	**-1.17**	-0.10	**-0.42**	**-1.43**	0.18	**0.52**	**1.00**
k	-0.09	-0.11	-0.17	-0.09	-0.11	-0.17	-0.09	-0.10	-0.16	**0.91**	**0.90**	**0.84**	0.12	0.12	0.19
q	0.10	0.16	**0.21**	0.10	0.16	**0.21**	0.10	0.15	**0.21**	0.09	0.15	**0.20**	-0.13	**-0.26**	**-0.46**
$\mu(c)$	0.14	**0.24**	**0.46**	0.12	**0.21**	**0.34**	0.08	0.12	0.14	0.04	0.05	-0.01	-0.16	**-0.34**	**-0.75**
$\sigma^2(c)$	**0.66**	**1.14**	**1.93**	**0.58**	**1.02**	**1.62**	**0.33**	**0.62**	**0.94**	0.12	**0.27**	**0.38**	**-0.96**	**-1.66**	**-2.64**

[a] Model responses are proportions of forest overstory in P1 (pine, <16 years), P2 (pine, 16-40 years), P3 (pine, 40-80 years), P4 (pine, ≥80 years), and UH (upland hardwood).

Table 5. Compartment permutation sequences selected for simulation experiment and associated inter-compartment distance statistics.

	Sequence								
Group	1	2	3	4	5	6	7	8	C[a]
1	5	13	4	9	9	13	8	11	1
	17	19	9	13	22	17	15	17	8
	21	26	16	17	24	27	19	23	13
	28	32	22	23	27	31	23	25	24
	32	34	27	27	33	33	27	33	32
2	15	4	3	1	5	1	3	2	9
	24	17	12	12	13	3	14	12	12
	29	21	23	20	17	6	24	20	21
	31	25	33	33	28	20	30	31	25
3	3	8	7	7	20	18	7	1	10
	6	11	14	18	30	23	10	8	27
	20	24	19	22	32	28	16	24	30
	25	27	26	31	34	30	29	30	33
4	1	14	1	3	3	4	6	4	6
	2	18	5	10	10	9	9	15	14
	7	22	20	16	16	16	21	26	23
	10	31	30	24	21	22	26	32	29
5	4	7	2	5	2	12	2	3	4
	13	10	10	6	7	24	11	10	7
	19	12	18	30	11	25	18	27	18
	27	16	24	34	31	32	33	29	22
6	8	1	6	11	4	5	4	9	3
	11	2	21	14	8	7	12	18	5
	14	9	25	19	23	10	28	19	15
	26	15	29	26	25	19	32	28	20
7	9	3	8	4	15	2	1	7	11
	12	5	13	21	18	8	25	13	17
	23	6	17	25	26	15	31	16	26
	33	29	32	32	29	34	34	22	34

Table 5. Continued.

	Sequence								
Group	1	2	3	4	5	6	7	8	C[a]
8	16	20	11	2	1	11	5	5	2
	18	23	15	8	6	14	13	6	16
	22	28	28	15	12	21	17	14	19
	30	30	31	28	14	26	20	21	28
	34	33	34	29	19	29	22	34	31
DBT (m)[b]	6989	7096	7732	7800	8092	8255	7276	7240	7489
rank (%)[b]	0.13	0.49	87.93	93.69	99.76	99.96	4.06	2.77	36.02
DWT (m)[c]	7175	6949	8601	8503	6953	7133	8651	8619	8118
rank (%)[c]	0.37	0.03	98.78	95.43	0.04	0.26	99.49	99.09	50.50

[a] Operational permutation sequence in current use by Piedmont National Wildlife Refuge.

[b] Average distance between compartments in successive years, and percent rank of average distance.

[c] Average distance between compartments within years, and percent rank of average distance.

Table 6. Split-plot analysis of variance for simulation outcomes, by model alternative, for 100-year total abundances of red-cockaded woodpecker (RCW) active clusters, wood thrush (WOTH, doubly log-transformed), and composite species score. Sources of variation were compartment burn limit (BURN), levels of average successive-year inter-compartment distance (DBT), levels of average within-year inter-compartment distance (DWT), and the DBT × DWT interaction.

			Response Variable					
			RCW		log-log(WOTH)		Composite[b]	
Model[a]	Source	df	MS	*P*	MS	*P*	MS	*P*
M_{000}	DBT	1	1386.11	0.104	0.030323	0.384	0.000050	0.990
	DWT	1	9.11	0.873	0.017469	0.500	0.116488	0.565
	DBT × DWT	1	1638.05	0.085	0.040881	0.320	0.845515	0.166
	error[c]	4	314.27	0.415	0.031771	0.451	0.296331	0.429
	BURN	1	519064.20	<0.001	6.852933	<0.001	3.564908	0.008
	error[d]	7	277.84	0.672	0.030602	0.259	0.271241	0.277
	residual	304	397.01		0.023869		0.217485	
M_{001}	DBT	1	1562.03	0.128	0.000062	0.820	0.118466	0.501
	DWT	1	50.40	0.748	0.000005	0.949	0.002732	0.916
	DBT × DWT	1	1762.50	0.112	0.001823	0.257	0.003132	0.910
	error[c]	4	426.60	0.348	0.001045	0.515	0.217188	0.515
	BURN	1	514483.00	<0.001	0.261219	<0.001	2.967860	0.010
	error[d]	7	321.65	0.555	0.001169	0.211	0.242949	0.365
	residual	304	383.09		0.000844		0.221641	

Table 6. Continued.

			Response Variable					
			RCW		log-log(WOTH)		Composite[b]	
Model[a]	Source	df	MS	*P*	MS	*P*	MS	*P*
M_{010}	DBT	1	2025.08	0.222	0.002556	0.650	0.363516	0.243
	DWT	1	222.78	0.657	0.013562	0.322	0.197783	0.369
	DBT × DWT	1	222.78	0.657	0.029340	0.172	0.335656	0.258
	error[c]	4	972.55	0.037	0.010644	0.782	0.193704	0.355
	BURN	1	596764.88	<0.001	6.783955	<0.001	3.327772	0.002
	error[d]	7	206.86	0.779	0.024629	0.327	0.148453	0.665
	residual	304	362.30		0.021273		0.209723	
M_{011}	DBT	1	994.05	0.226	0.002353	0.103	0.956877	0.124
	DWT	1	30.01	0.816	0.000024	0.842	0.015326	0.818
	DBT × DWT	1	1224.61	0.188	0.001224	0.203	0.704320	0.171
	error[c]	4	486.71	0.432	0.000531	0.449	0.254287	0.411
	BURN	1	561795.20	<0.001	0.242493	<0.001	3.016155	0.008
	error[d]	7	448.21	0.312	0.000509	0.687	0.222910	0.488
	residual	304	378.79		0.000745		0.241157	

Table 6. Continued.

			Response Variable					
			RCW		log-log(WOTH)		Composite[b]	
Model[a]	Source	df	MS	*P*	MS	*P*	MS	*P*
M_{100}	DBT	1	858.05	0.413	0.426130	0.184	1.288760	0.183
	DWT	1	1891.51	0.247	0.334553	0.228	1.485318	0.159
	DBT × DWT	1	238.05	0.656	0.027241	0.706	0.147768	0.615
	error[c]	4	1031.88	0.118	0.165689	0.424	0.498725	0.342
	BURN	1	465735.20	<0.001	36.248289	<0.001	1.192793	0.116
	error[d]	7	379.38	0.111	0.149696	0.009	0.370272	0.012
	residual	304	224.69		0.054750		0.140197	
M_{101}	DBT	1	23.65	0.799	0.003960	0.067	0.902022	0.061
	DWT	1	1197.38	0.124	0.000038	0.818	0.259079	0.239
	DBT × DWT	1	178.50	0.495	0.000069	0.757	0.085051	0.473
	error[c]	4	317.93	0.529	0.000634	0.450	0.135772	0.821
	BURN	1	499043.03	<0.001	0.151793	<0.001	4.534170	0.010
	error[d]	7	367.48	0.064	0.000609	0.314	0.364130	0.077
	residual	304	189.97		0.000516		0.196235	

Table 6. Continued.

			Response Variable					
			RCW		log-log(WOTH)		Composite[b]	
Model[a]	Source	df	MS	*P*	MS	*P*	MS	*P*
M_{110}	DBT	1	359.13	0.565	0.205997	0.185	0.600558	0.260
	DWT	1	948.75	0.366	0.000455	0.944	0.118156	0.592
	DBT × DWT	1	279.38	0.610	0.006527	0.790	0.010612	0.870
	error[c]	4	914.43	0.019	0.080446	0.579	0.349847	0.358
	BURN	1	479647.88	<0.001	38.151555	<0.001	0.730445	0.144
	error[d]	7	149.10	0.640	0.104905	0.063	0.269930	0.051
	residual	304	202.21		0.054039		0.132932	
M_{111}	DBT	1	0.00	0.999	0.001443	0.437	0.406274	0.559
	DWT	1	306.15	0.588	0.001415	0.441	0.166222	0.705
	DBT × DWT	1	630.00	0.447	0.000001	0.983	0.091809	0.777
	error[c]	4	886.04	0.045	0.001939	0.009	1.001722	0.001
	BURN	1	419847.75	<0.001	0.146190	<0.001	3.389662	<0.001
	error[d]	7	204.40	0.399	0.000233	0.710	0.065808	0.901
	residual	304	195.41		0.000356		0.163908	

Table 6. Continued.

			Response Variable					
			RCW		log-log(WOTH)		Composite[b]	
Model[a]	Source	df	MS	*P*	MS	*P*	MS	*P*
M_{200}	DBT	1	1252.15	0.024	0.000111	0.923	0.080885	0.699
	DWT	1	1615.50	0.016	0.002984	0.622	0.593085	0.323
	DBT × DWT	1	4039.90	0.003	0.003900	0.575	0.047887	0.765
	error[c]	4	99.27	0.993	0.010497	0.485	0.468409	0.785
	BURN	1	550207.38	<0.001	0.002031	0.679	51.025950	<0.001
	error[d]	7	1785.66	0.011	0.010935	0.053	1.095490	0.007
	residual	304	672.46		0.005418		0.388344	
M_{201}	DBT	1	21.01	0.844	0.000563	0.652	0.134452	0.696
	DWT	1	1155.20	0.195	0.000928	0.566	0.594762	0.426
	DBT × DWT	1	1970.11	0.112	0.000502	0.670	0.600147	0.424
	error[c]	4	477.73	0.636	0.002377	0.221	0.759168	0.231
	BURN	1	491097.80	<0.001	0.015397	0.010	34.025086	<0.001
	error[d]	7	718.12	0.318	0.001271	0.484	0.418973	0.501
	residual	304	612.53		0.001368		0.461672	

Table 6. Continued.

			Response Variable					
			RCW		log-log(WOTH)		Composite[b]	
Model[a]	Source	df	MS	*P*	MS	*P*	MS	*P*
M_{210}	DBT	1	51.20	0.767	0.003179	0.436	0.135456	0.507
	DWT	1	1022.45	0.229	0.004406	0.366	0.687145	0.176
	DBT × DWT	1	762.61	0.287	0.001227	0.619	0.000003	0.997
	error[c]	4	507.02	0.578	0.004245	0.166	0.255488	0.471
	BURN	1	596160.45	<0.001	0.007547	0.086	47.424564	<0.001
	error[d]	7	659.19	0.411	0.001899	0.876	0.257593	0.747
	residual	304	641.34		0.004305		0.421786	
M_{211}	DBT	1	165.31	0.528	0.000485	0.484	0.026068	0.765
	DWT	1	3604.61	0.032	0.006330	0.050	3.056891	0.026
	DBT × DWT	1	3.61	0.924	0.000013	0.907	0.000773	0.959
	error[c]	4	347.16	0.916	0.000817	0.633	0.254870	0.840
	BURN	1	455718.05	<0.001	0.017074	0.007	31.308462	<0.001
	error[d]	7	1539.74	0.013	0.001219	0.452	0.741135	0.108
	residual	304	596.14		0.001254		0.435578	

[a] Alternative models are combinations of hypotheses regarding creation and persistence of old pine habitat (subscript 1; 0 = moderate persistence, 1 = low persistence, 2 = high persistence), sensitivity of woodpecker productivity to amount of foraging habitat (subscript 2; 0 = insensitive, 1 = sensitive), and linearity of wood thrush response to habitat conditions (subscript 3; 0 = linear, 1 = nonlinear).

Table 6. Continued.

[b] Average of outcomes for total active red-cockaded woodpecker clusters and for log-log(wood thrush abundance), each component scaled by its within-model standard deviation.

[c] Pooled variance among permutation sequence means within combinations of DWT and DBT. Low *P* for this source suggests variance within DWT-DBT combinations was larger than variance within BURN-DWT-DBT combinations. This source serves as the error mean square for the main-plot effects (DBT and DWT) and interaction (DBT × DWT).

[d] Pooled variance among burn level means within combinations of BURN, DWT, and DBT. Low *P* for this source suggests variance within BURN-DWT-DBT combinations was larger than sequence replication variance. This source serves as the error mean square for the BURN split-plot effect.

Table 7. Mean simulation outcomes and standard error estimates of 100-year total abundances of red-cockaded woodpecker (RCW) active clusters, wood thrush (WOTH, doubly log-transformed), and composite species score, by simulation model alternative and decision class level. Decision classes were levels of average successive-year inter-compartment distance (DBT, low vs. high), levels of average within-year inter-compartment distance (DWT, low vs. high), and compartment burn limits (BURN, 7 vs. 17).

			RCW		log-log(WOTH)		Composite[b]	
Model[a]	Effect	Level	$\bar{x}$	SE	$\bar{x}$	SE	$\bar{x}$	SE
M_{000}	DBT	Low	126.97	1.401	2.243	0.0141	6.631	0.0430
		High	122.81		2.262		6.630	
	DWT	Low	125.06	1.401	2.260	0.0141	6.650	0.0430
		High	124.72		2.245		6.612	
	DBT × DWT	Lo/Lo	124.88	1.982	2.239	0.0199	6.599	0.0609
		Lo/Hi	129.06		2.247		6.663	
		Hi/Lo	125.24		2.281		6.701	
		Hi/Hi	120.38		2.244		6.560	
	BURN	7	84.61	1.318	2.399	0.0138	6.525	0.0412
		17	165.16		2.106		6.736	
	overall mean		124.89	0.951	2.253	0.0069	6.631	0.0184
M_{001}	DBT	Low	126.36	1.633	1.995	0.0026	25.888	0.0368
		High	121.94		1.996		25.850	
	DWT	Low	124.55	1.633	1.995	0.0026	25.872	0.0368
		High	123.76		1.995		25.866	
	DBT × DWT	Lo/Lo	124.41	2.309	1.997	0.0036	25.894	0.0521
		Lo/Hi	128.31		1.993		25.882	
		Hi/Lo	124.69		1.993		25.850	
		Hi/Hi	119.20		1.998		25.850	
	BURN	7	84.06	1.418	2.024	0.0027	25.773	0.0390
		17	164.25		1.967		25.966	
	overall model mean		124.15	0.780	1.995	0.0015	25.869	0.0204

Table 7. Continued

			RCW		log-log(WOTH)		Composite[b]	
Model[a]	Effect	Level	$\bar{x}$	SE	$\bar{x}$	SE	$\bar{x}$	SE
M_{010}	DBT	Low	125.97	2.465	2.240	0.0082	6.841	0.0348
		High	120.94		2.235		6.773	
	DWT	Low	122.62	2.465	2.231	0.0082	6.782	0.0348
		High	124.29		2.244		6.832	
	DBT × DWT	Lo/Lo	124.30	3.487	2.224	0.0115	6.784	0.0492
		Lo/Hi	127.64		2.256		6.898	
		Hi/Lo	120.94		2.238		6.781	
		Hi/Hi	120.94		2.232		6.766	
	BURN	7	80.27	1.137	2.383	0.0124	6.705	0.0305
		17	166.64		2.092		6.909	
	overall model mean		123.45	0.588	2.238	0.0108	6.807	0.0264
M_{011}	DBT	Low	125.34	1.744	1.998	0.0018	27.495	0.0399
		High	121.81		1.993		27.385	
	DWT	Low	123.88	1.744	1.996	0.0018	27.447	0.0399
		High	123.27		1.995		27.433	
	DBT × DWT	Lo/Lo	123.69	2.467	1.997	0.0026	27.455	0.0564
		Lo/Hi	126.99		2.000		27.535	
		Hi/Lo	124.08		1.995		27.439	
		Hi/Hi	119.55		1.991		27.331	
	BURN	7	81.68	1.674	2.023	0.0018	27.343	0.0373
		17	165.48		1.968		27.537	
	overall model mean		123.58	1.302	1.995	0.0012	27.440	0.0265

Table 7. Continued

			RCW		log-log(WOTH)		Composite[b]	
Model[a]	Effect	Level	$\bar{x}$	SE	$\bar{x}$	SE	$\bar{x}$	SE
M_{100}	DBT	Low	105.47	2.540	2.583	0.0322	4.365	0.0558
		High	108.74		2.656		4.492	
	DWT	Low	104.68	2.540	2.587	0.0322	4.360	0.0558
		High	109.54		2.651		4.496	
	DBT × DWT	Lo/Lo	102.18	3.591	2.541	0.0455	4.275	0.0790
		Lo/Hi	108.76		2.624		4.454	
		Hi/Lo	107.18		2.632		4.445	
		Hi/Hi	110.31		2.679		4.538	
	BURN	7	68.96	1.540	2.956	0.0306	4.367	0.0481
		17	145.26		2.282		4.489	
	overall model mean		107.11	1.001	2.619	0.0189	4.428	0.0264
M_{101}	DBT	Low	109.43	1.410	2.042	0.0020	33.989	0.0291
		High	109.98		2.035		33.883	
	DWT	Low	107.77	1.410	2.039	0.0020	33.908	0.0291
		High	111.64		2.039		33.965	
	DBT × DWT	Lo/Lo	106.75	1.994	2.042	0.0028	33.944	0.0412
		Lo/Hi	112.11		2.043		34.034	
		Hi/Lo	108.79		2.036		33.871	
		Hi/Hi	111.16		2.035		33.895	
	BURN	7	70.21	1.515	2.061	0.0020	33.817	0.0477
		17	149.19		2.017		34.055	
	overall model mean		109.70	1.037	2.039	0.0014	33.936	0.0332

Table 7. Continued

			RCW		log-log(WOTH)		Composite[b]	
Model[a]	Effect	Level	$\bar{x}$	SE	$\bar{x}$	SE	$\bar{x}$	SE
M_{110}	DBT	Low	105.71	2.391	2.603	0.0224	4.409	0.0468
		High	107.83		2.654		4.496	
	DWT	Low	105.04	2.391	2.630	0.0224	4.433	0.0468
		High	108.49		2.628		4.472	
	DBT × DWT	Lo/Lo	103.05	3.381	2.609	0.0317	4.384	0.0661
		Lo/Hi	108.36		2.598		4.434	
		Hi/Lo	107.04		2.651		4.482	
		Hi/Hi	108.61		2.657		4.509	
	BURN	7	68.05	0.965	2.974	0.0256	4.405	0.0411
		17	145.48		2.283		4.500	
	overall model mean		106.77	0.656	2.629	0.0158	4.452	0.0253
M_{111}	DBT	Low	105.93	2.353	2.044	0.0035	35.604	0.0791
		High	105.93		2.039		35.533	
	DWT	Low	106.91	2.353	2.040	0.0035	35.546	0.0791
		High	104.95		2.044		35.591	
	DBT × DWT	Lo/Lo	108.31	3.328	2.042	0.0049	35.598	0.1119
		Lo/Hi	103.55		2.046		35.610	
		Hi/Lo	105.50		2.037		35.493	
		Hi/Hi	106.35		2.042		35.573	
	BURN	7	69.71	1.130	2.063	0.0012	35.466	0.0203
		17	142.15		2.020		35.672	
	overall model mean		105.93	0.578	2.042	0.0010	35.569	0.0158

Table 7. Continued

			RCW		log-log(WOTH)		Composite[b]	
Model[a]	Effect	Level	$\bar{x}$	SE	$\bar{x}$	SE	$\bar{x}$	SE
M_{200}	DBT	Low	137.61	0.788	2.014	0.0081	14.898	0.0541
		High	141.57		2.013		14.930	
	DWT	Low	137.34	0.788	2.010	0.0081	14.871	0.0541
		High	141.84		2.016		14.957	
	DBT × DWT	Lo/Lo	131.81	1.114	2.014	0.0115	14.842	0.0765
		Lo/Hi	143.41		2.013		14.953	
		Hi/Lo	142.88		2.006		14.899	
		Hi/Hi	140.26		2.019		14.960	
	BURN	7	98.13	3.341	2.016	0.0083	14.514	0.0827
		17	181.06		2.011		15.313	
	overall model mean		139.59	2.240	2.013	0.0062	14.914	0.0578
M_{201}	DBT	Low	138.66	1.728	1.950	0.0039	27.578	0.0689
		High	138.14		1.947		27.537	
	DWT	Low	136.50	1.728	1.947	0.0039	27.514	0.0689
		High	140.30		1.950		27.601	
	DBT × DWT	Lo/Lo	134.28	2.444	1.947	0.0055	27.492	0.0974
		Lo/Hi	143.04		1.953		27.664	
		Hi/Lo	138.73		1.946		27.537	
		Hi/Hi	137.56		1.947		27.537	
	BURN	7	99.23	2.119	1.955	0.0028	27.231	0.0512
		17	177.58		1.941		27.884	
	overall model mean		138.40	1.367	1.948	0.0026	27.557	0.0454

Table 7. Continued

			RCW		log-log(WOTH)		Composite[b]	
Model[a]	Effect	Level	$\bar{x}$	SE	$\bar{x}$	SE	$\bar{x}$	SE
M_{210}	DBT	Low	140.77	1.780	2.016	0.0052	17.047	0.0400
		High	139.97		2.023		17.088	
	DWT	Low	138.58	1.780	2.016	0.0052	17.021	0.0400
		High	142.16		2.023		17.114	
	DBT × DWT	Lo/Lo	137.44	2.517	2.015	0.0073	17.001	0.0565
		Lo/Hi	144.10		2.018		17.093	
		Hi/Lo	139.73		2.017		17.042	
		Hi/Hi	140.21		2.028		17.134	
	BURN	7	97.21	2.030	2.024	0.0034	16.682	0.0401
		17	183.53		2.015		17.452	
	overall model mean		140.37	1.688	2.019	0.0026	17.067	0.0302
M_{211}	DBT	Low	136.01	1.473	1.947	0.0023	28.202	0.0399
		High	134.58		1.950		28.220	
	DWT	Low	131.94	1.473	1.944	0.0023	28.113	0.0399
		High	138.65		1.953		28.309	
	DBT × DWT	Lo/Lo	132.55	2.083	1.943	0.0032	28.106	0.0564
		Lo/Hi	139.48		1.952		28.298	
		Hi/Lo	131.33		1.945		28.121	
		Hi/Hi	137.83		1.955		28.319	
	BURN	7	97.56	3.102	1.956	0.0028	27.898	0.0681
		17	173.03		1.941		28.524	
	overall model mean		135.29	2.158	1.949	0.0018	28.211	0.0455

[a] Alternative models are combinations of hypotheses regarding creation and persistence of old pine habitat (subscript 1; 0 = moderate persistence, 1 = low persistence, 2 = high persistence), sensitivity of woodpecker productivity to amount of foraging habitat (subscript 2; 0 = insensitive, 1 = sensitive), and linearity of wood thrush response to habitat conditions (subscript 3; 0 = linear, 1 = nonlinear).

[b] Average of outcomes for total active red-cockaded woodpecker clusters and for log-log(wood thrush abundance), each component scaled by its within-model standard deviation.

Table 8. Mean simulation outcomes and ranks, by simulation model alternative and averaged over all models, of 100-year total abundance of red-cockaded woodpecker active clusters. Outcomes were averaged over both permutation sequences within a decision class. Decision classes were combinations of compartment burn limits (BURN, 7 vs. 17), levels of average successive-year inter-compartment distance (DBT, Low vs. High), and levels of average within-year inter-compartment distance (DWT, Low vs. High). Estimates of standard error were based on between-sequence variation within decision classes. Means for operational PNWR permutation sequences (DBT = "C", DWT = "C") are also presented and ranked.

			Model[a]								
			M_{000}		M_{001}		M_{010}		M_{011}		
BURN	DBT	DWT	$\bar{x}$	rank	$\bar{x}$	rank	$\bar{x}$	rank	$\bar{x}$	rank	(continued)
7	L	L	86.30	7	81.63	8	80.18	7	82.85	8	
	L	H	87.08	6	86.70	6	82.18	6	82.20	9	
	H	L	83.55	8	86.58	7	80.18	7	83.78	7	
	H	H	81.53	9	81.33	9	78.55	9	77.88	10	
	C	C	81.25	10	79.35	10	78.15	10	88.05	6	
17	L	L	163.45	4	167.20	2	168.43	2	164.53	3	
	L	H	171.05	1	169.93	1	173.10	1	171.78	1	
	H	L	166.93	3	162.80	4	161.70	4	164.38	4	
	H	H	159.23	5	157.08	5	163.33	3	161.23	5	
	C	C	168.20	2	164.25	3	159.40	5	164.70	2	
SE for means			2.689		2.206		1.662		3.684		

Table 8. Continued.

			Model[a]								
			M_{100}		M_{101}		M_{110}		M_{111}		
BURN	DBT	DWT	$\bar{x}$	rank	$\bar{x}$	rank	$\bar{x}$	rank	$\bar{x}$	rank	(continued)
7	L	L	62.03	10	66.33	9	64.05	9	69.15	8	
	L	H	69.35	8	71.68	7	67.95	8	67.85	9	
	H	L	72.43	6	72.63	6	70.10	6	71.08	6	
	H	H	72.03	7	70.23	8	70.10	6	70.75	7	
	C	C	63.70	9	64.75	10	62.75	10	64.10	10	
17	L	L	142.33	3	147.18	4	142.05	5	147.48	1	
	L	H	148.18	2	152.55	1	148.78	1	139.25	5	
	H	L	141.93	4	144.95	5	143.98	4	139.93	4	
	H	H	148.60	1	152.10	2	147.13	2	141.95	3	
	C	C	137.80	5	149.00	3	145.20	3	142.15	2	
SE for means			2.830		2.934		1.855		1.636		

Table 8. Continued.

			Model[a]								Model average	
			M_{200}		M_{201}		M_{210}		M_{211}			
BURN	DBT	DWT	$\bar{x}$	rank	$\bar{x}$	rank	$\bar{x}$	rank	$\bar{x}$	rank	$\bar{x}$	rank
7	L	L	89.25	10	96.25	9	93.58	9	89.30	9	80.07	9
	L	H	95.80	8	99.05	8	99.63	6	100.20	7	84.14	7
	H	L	107.35	6	100.45	7	99.55	7	95.98	8	85.30	6
	H	H	100.10	7	101.15	6	96.08	8	104.75	6	83.70	8
	C	C	90.40	9	85.95	10	85.30	10	88.65	10	77.70	10
17	L	L	174.38	5	172.30	4	181.30	4	175.80	3	162.20	3
	L	H	191.03	1	187.03	1	188.58	1	178.75	1	168.33	1
	H	L	178.40	4	177.00	2	179.90	5	166.68	5	160.71	5
	H	H	180.43	3	173.98	3	184.35	3	170.90	4	161.69	4
	C	C	184.60	2	169.15	5	186.80	2	176.85	2	162.34	2
SE for means			6.335		3.867		4.774		6.105			

[a] Alternative models are combinations of hypotheses regarding creation and persistence of old pine habitat (subscript 1; 0 = moderate persistence, 1 = low persistence, 2 = high persistence), sensitivity of woodpecker productivity to amount of foraging habitat (subscript 2; 0 = insensitive, 1 = sensitive), and linearity of wood thrush response to habitat conditions (subscript 3; 0 = linear, 1 = nonlinear).

Table 9. Mean simulation outcomes and ranks, by simulation model alternative and averaged over all models, of 100-year total abundances of wood thrush (doubly log-transformed). Outcomes were averaged over both permutation sequences within a decision class. Decision classes were combinations of compartment burn limits (BURN, 7 vs. 17), levels of average successive-year inter-compartment distance (DBT, Low vs. High), and levels of average within-year inter-compartment distance (DWT, Low vs. High). Estimates of standard error were based on between-sequence variation within decision classes. Means for operational PNWR permutation sequences (DBT = "C", DWT = "C") are also presented and ranked.

			Model[a]								
			M_{000}		M_{001}		M_{010}		M_{011}		
BURN	DBT	DWT	$\bar{x}$	rank	$\bar{x}$	rank	$\bar{x}$	rank	$\bar{x}$	rank	(continued)
7	L	L	2.397	3	2.029	1	2.372	3	2.026	2	
	L	H	2.374	5	2.014	5	2.403	1	2.025	3	
	H	L	2.405	2	2.024	4	2.370	4	2.025	4	
	H	H	2.420	1	2.029	2	2.388	2	2.016	5	
	C	C	2.397	4	2.027	3	2.357	5	2.027	1	
17	L	L	2.080	8	1.965	9	2.077	8	1.967	8	
	L	H	2.120	7	1.971	6	2.110	6	1.974	6	
	H	L	2.157	6	1.963	10	2.105	7	1.965	10	
	H	H	2.067	9	1.968	8	2.075	9	1.965	9	
	C	C	2.042	10	1.970	7	2.073	10	1.972	7	
SE for means			0.0194		0.0043		0.0305		0.0034		

Table 9. Continued.

			Model[a]								
			M_{100}		M_{101}		M_{110}		M_{111}		
BURN	DBT	DWT	$\bar{x}$	rank	$\bar{x}$	rank	$\bar{x}$	rank	$\bar{x}$	rank	(continued)
7	L	L	2.811	5	2.062	2	2.960	3	2.064	3	
	L	H	2.951	4	2.063	1	2.885	4	2.067	1	
	H	L	3.011	3	2.061	3	3.037	1	2.057	5	
	H	H	3.049	1	2.056	5	3.014	2	2.064	2	
	C	C	3.015	2	2.057	4	2.829	5	2.059	4	
17	L	L	2.271	8	2.021	7	2.258	9	2.020	7	
	L	H	2.297	7	2.023	6	2.310	6	2.025	6	
	H	L	2.254	10	2.010	10	2.265	8	2.018	9	
	H	H	2.308	6	2.015	8	2.300	7	2.019	8	
	C	C	2.270	9	2.014	9	2.258	10	2.003	10	
SE for means			0.0535		0.0040		0.0447		0.0027		

Table 9. Continued.

			Model[a]								Model average	
			M_{200}		M_{201}		M_{210}		M_{211}			
BURN	DBT	DWT	$\bar{x}$	rank	$\bar{x}$	rank	$\bar{x}$	rank	$\bar{x}$	rank	$\bar{x}$	rank
7	L	L	2.007	8	1.953	5	2.021	6	1.946	4	2.221	5
	L	H	2.007	7	1.960	1	2.022	5	1.959	2	2.228	3
	H	L	2.021	3	1.955	2	2.027	3	1.951	3	2.245	2
	H	H	2.027	1	1.954	3	2.027	2	1.968	1	2.251	1
	C	C	2.014	5	1.953	4	2.013	8	1.936	10	2.224	4
17	L	L	2.021	2	1.940	8	2.008	9	1.940	8	2.047	9
	L	H	2.019	4	1.946	6	2.014	7	1.944	5	2.063	6
	H	L	1.991	10	1.938	10	2.007	10	1.939	9	2.051	8
	H	H	2.011	6	1.941	7	2.029	1	1.941	7	2.053	7
	C	C	2.006	9	1.938	9	2.025	4	1.943	6	2.043	10
SE for means			0.0175		0.0074		0.0073		0.0051			

[a] Alternative models are combinations of hypotheses regarding creation and persistence of old pine habitat (subscript 1; 0 = moderate persistence, 1 = low persistence, 2 = high persistence), sensitivity of woodpecker productivity to amount of foraging habitat (subscript 2; 0 = insensitive, 1 = sensitive), and linearity of wood thrush response to habitat conditions (subscript 3; 0 = linear, 1 = nonlinear).

Table 10. Mean simulation outcomes and ranks, by simulation model alternative and averaged over all models, of 100-year total values of composite species average[a]. Outcomes were averaged over both permutation sequences within a decision class. Decision classes were combinations of compartment burn limits (BURN, 7 vs. 17), levels of average successive-year inter-compartment distance (DBT, Low vs. High), and levels of average within-year inter-compartment distance (DWT, Low vs. High). Estimates of standard error were based on between-sequence variation within decision classes. Means for operational PNWR permutation sequences (DBT = “C”, DWT = “C”) are also presented and ranked.

			Model[b]								
			M_{000}		M_{001}		M_{010}		M_{011}		
BURN	DBT	DWT	$\bar{x}$	rank	$\bar{x}$	rank	$\bar{x}$	rank	$\bar{x}$	rank	(continued)
7	L	L	6.540	6	25.810	6	6.676	8	27.397	7	
	L	H	6.494	9	25.684	10	6.774	6	27.379	9	
	H	L	6.527	8	25.796	8	6.672	9	27.391	8	
	H	H	6.539	7	25.801	7	6.699	7	27.205	10	
	C	C	6.484	10	25.764	9	6.618	10	27.469	5	
17	L	L	6.657	3	25.979	3	6.891	2	27.512	3	
	L	H	6.833	2	26.080	1	7.023	1	27.690	1	
	H	L	6.875	1	25.903	4	6.890	3	27.487	4	
	H	H	6.580	5	25.899	5	6.833	4	27.458	6	
	C	C	6.621	4	26.007	2	6.787	5	27.578	2	
SE for means			0.0522		0.0578		0.0747		0.0749		

Table 10. Continued.

			Model^b								
			M_{100}		M_{101}		M_{110}		M_{111}		
BURN	DBT	DWT	$\bar{x}$	rank	$\bar{x}$	rank	$\bar{x}$	rank	$\bar{x}$	rank	(continued)
7	L	L	4.111	10	33.794	8	4.339	8	35.468	7	
	L	H	4.367	9	33.879	6	4.296	9	35.512	5	
	H	L	4.475	4	33.855	7	4.505	3	35.382	8	
	H	H	4.516	3	33.740	9	4.478	4	35.500	6	
	C	C	4.375	8	33.689	10	4.166	10	35.325	10	
17	L	L	4.440	5	34.095	2	4.429	7	35.728	1	
	L	H	4.542	2	34.189	1	4.572	1	35.708	2	
	H	L	4.415	6	33.887	5	4.460	6	35.605	4	
	H	H	4.561	1	34.050	3	4.540	2	35.645	3	
	C	C	4.384	7	33.997	4	4.467	5	35.378	9	
SE for means			0.0748		0.0939		0.0715		0.0447		

Table 10. Continued.

			Model[b]								Model average	
			M_{200}		M_{201}		M_{210}		M_{211}			
BURN	DBT	DWT	$\bar{x}$	rank	$\bar{x}$	rank	$\bar{x}$	rank	$\bar{x}$	rank	$\bar{x}$	rank
7	L	L	14.368	10	27.170	9	16.622	9	27.672	9	19.164	9
	L	H	14.436	8	27.288	6	16.688	8	27.969	7	19.230	8
	H	L	14.642	6	27.239	7	16.724	6	27.816	8	19.252	7
	H	H	14.612	7	27.229	8	16.696	7	28.136	6	19.263	6
	C	C	14.426	9	27.065	10	16.479	10	27.532	10	19.116	10
17	L	L	15.317	3	27.813	4	17.379	4	28.540	3	19.565	3
	L	H	15.470	1	28.041	1	17.499	3	28.628	1	19.690	1
	H	L	15.155	5	27.835	3	17.359	5	28.425	5	19.525	5
	H	H	15.309	4	27.844	2	17.573	1	28.503	4	19.566	2
	C	C	15.320	2	27.752	5	17.561	2	28.594	2	19.537	4
SE for means			0.1634		0.1284		0.0855		0.1288			

[a] Average of outcomes for total active red-cockaded woodpecker clusters and for log-log(wood thrush abundance), each component scaled by its within-model standard deviation.

[b] Alternative models are combinations of hypotheses regarding creation and persistence of old pine habitat (subscript 1; 0 = moderate persistence, 1 = low persistence, 2 = high persistence), sensitivity of woodpecker productivity to amount of foraging habitat (subscript 2; 0 = insensitive, 1 = sensitive), and linearity of wood thrush response to habitat conditions (subscript 3; 0 = linear, 1 = nonlinear).

Table 11. Abundance of red-cockaded woodpecker active clusters on Piedmont National Wildlife Refuge in 2001, by Refuge compartment, and mean and range of average predicted values of 2001 cluster abundance for twelve alternative forest simulation models.

		Average Predicted Abundance	
Compartment	Survey	$\bar{x}$	range
1	0	0.000	0.000 - 0.000
2	6	5.976	5.953 - 5.994
3	5	4.980	4.962 - 4.994
4	1	0.995	0.990 - 0.999
5	0	0.000	0.000 - 0.000
6	2	1.993	1.985 - 1.999
7	1	1.986	1.974 - 1.994
8	2	1.980	1.975 - 1.987
9	0	1.948	1.935 - 1.963
10	2	1.065	1.042 - 1.082
11	0	0.000	0.000 - 0.000
12	0	0.000	0.000 - 0.000
13	0	0.961	0.951 - 0.968
14	2	2.860	2.824 - 2.885
15	2	1.985	1.972 - 1.993
16	2	2.365	2.346 - 2.380
17	1	0.980	0.970 - 0.990
18	0	0.000	0.000 - 0.000
19	2	1.992	1.978 - 1.998
20	0	0.000	0.000 - 0.000
21	3	1.859	1.838 - 1.894
22	3	2.988	2.975 - 2.997
23	1	0.996	0.991 - 0.999
24	4	1.536	1.504 - 1.554
25	0	0.886	0.868 - 0.912
26	0	0.000	0.000 - 0.000
27	0	0.000	0.000 - 0.000
28	0	0.634	0.613 - 0.671
29	0	0.388	0.354 - 0.409
30	0	0.425	0.395 - 0.446
31	0	0.000	0.000 - 0.000
32	0	0.000	0.000 - 0.000
33	0	0.000	0.000 - 0.000

Table 11. Continued.

Compartment	Survey	Average Predicted Abundance $\bar{x}$	range
34	0	0.994	0.989 - 1.000
Total	39	42.772	42.567 - 42.926

Table 12. Likelihood values ($\mathscr{L}$) and posterior probabilities (p_1), conditional on year 2001 observed abundances of active woodpecker clusters and prior probabilities (p_0), for alternative forest and bird simulation models.

Model	p_0	$\mathscr{L}$	p_1
M_{000}	0.0833	8.74×10^{-14}	0.0808
M_{001}	0.0833	8.87×10^{-14}	0.0820
M_{010}	0.0833	9.17×10^{-14}	0.0848
M_{011}	0.0833	9.15×10^{-14}	0.0846
M_{100}	0.0833	8.34×10^{-14}	0.0771
M_{101}	0.0833	9.47×10^{-14}	0.0876
M_{110}	0.0833	9.28×10^{-14}	0.0858
M_{111}	0.0833	9.69×10^{-14}	0.0896
M_{200}	0.0833	9.00×10^{-14}	0.0832
M_{201}	0.0833	8.66×10^{-14}	0.0801
M_{210}	0.0833	8.88×10^{-14}	0.0821
M_{211}	0.0833	8.90×10^{-14}	0.0823

APPENDIX B

FIGURES

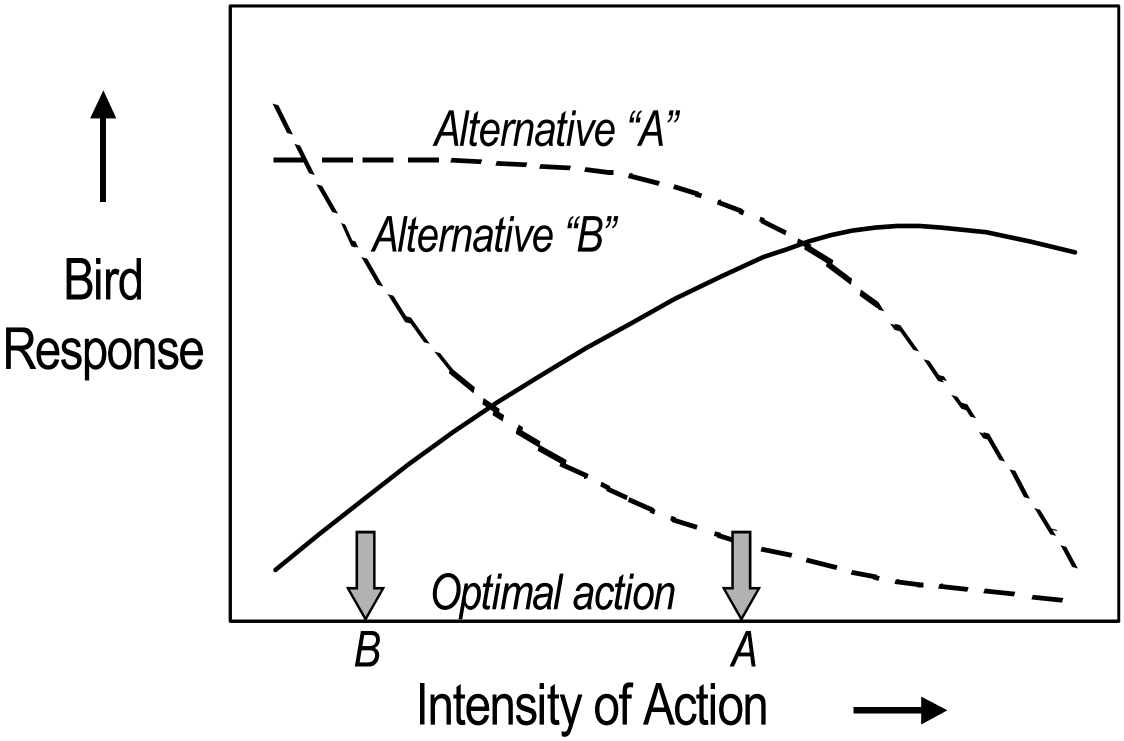


Figure 1. Uncertainty in resource response implies an unknown degree of trade-off among multiple resource objectives. Hypothetical responses by two bird species to a management action are displayed, but response by one species is uncertain and two alternatives for its response are represented by dashed lines. If the objective of management is to maximize a composite response of the two species, then the optimal action under each uncertain alternative is indicated by a shaded arrow. If the true response by the bird species is alternative A, then management trade-off between the two species is slight, as responses by both bird populations at arrow A are near their maxima. However, if the true response is alternative B, then management trade-off is more severe.

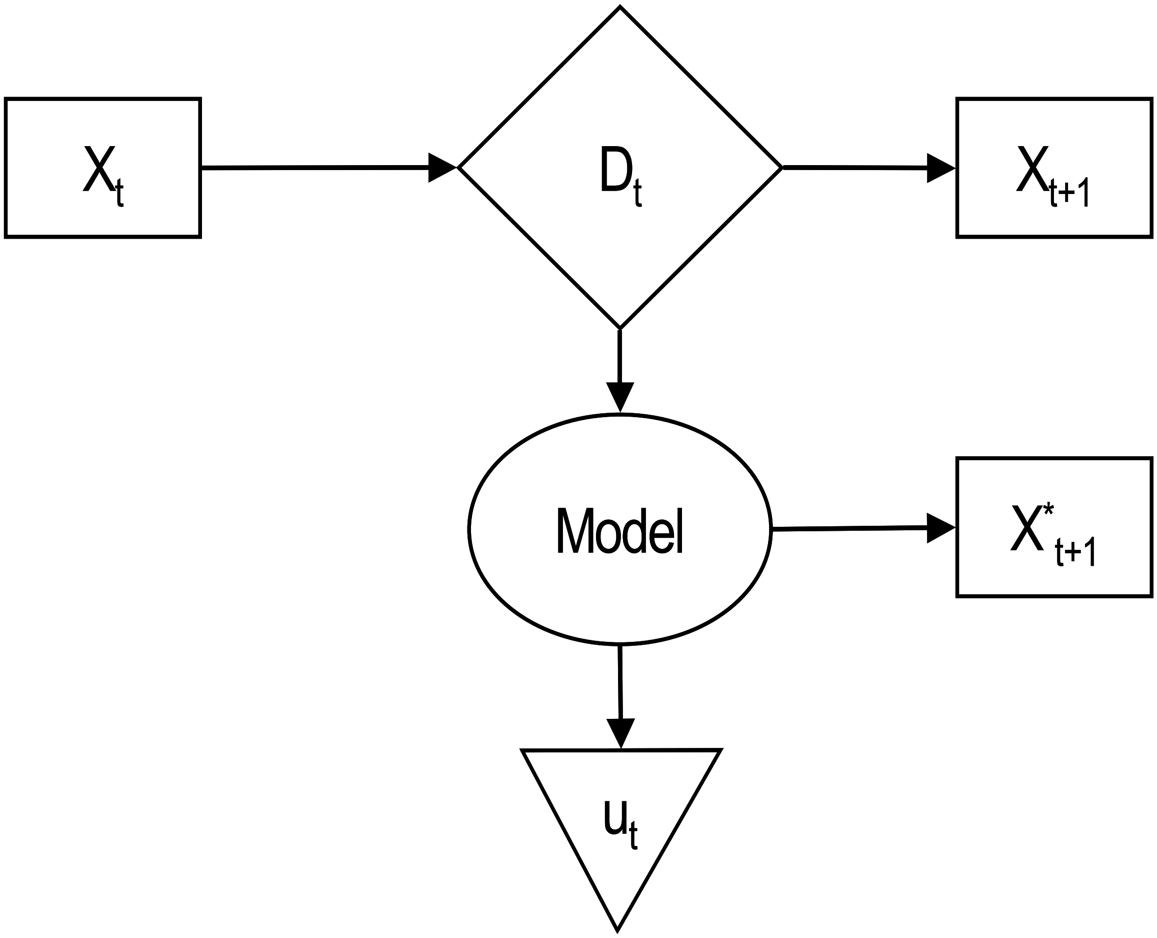


Figure 2. Diagram of a dynamic decision making process. The resource state, *X*, is advanced from time *t* to time *t*+1 through a management decision, *D*, made at time *t*. Each decision in the decision set generates a reward, and the value of the reward, *u*, is determined by a model. The model also provides a prediction X^* of the state of the system at time *t*+1. The decision cycle then repeats at time *t*+1.

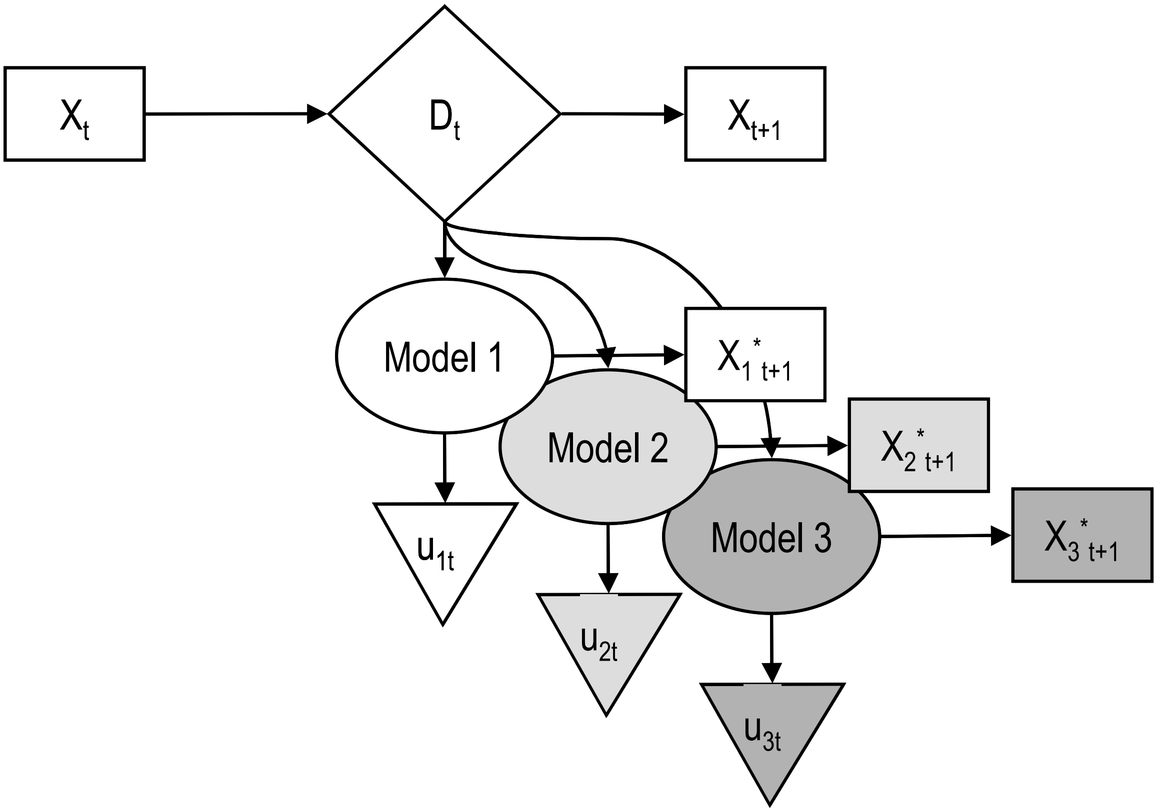


Figure 3. Structural uncertainty in decision making implies a choice among multiple, plausible system models. Selection of the optimal decision and the forecast of the future system state depends upon the model in which the decision maker places his belief.

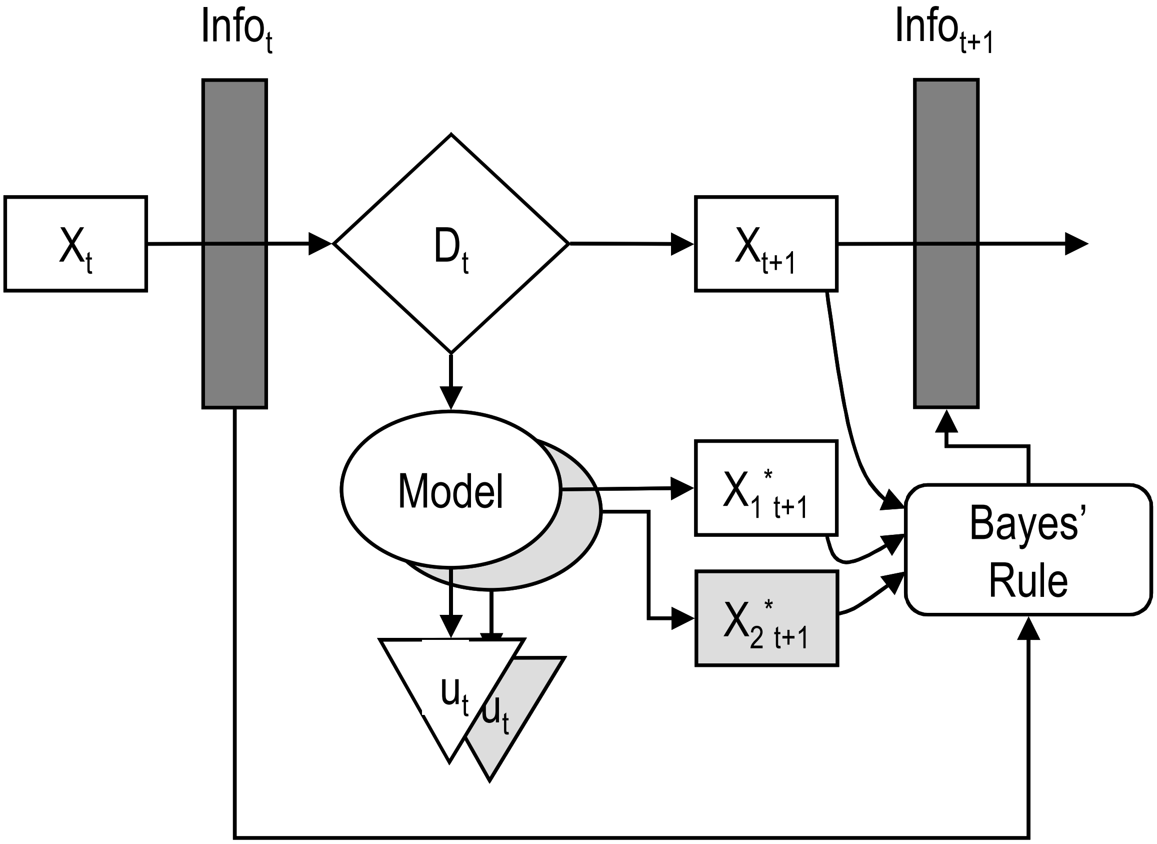


Figure 4. Cycle of decision making, monitoring, and information updating that characterizes adaptive management. Given the resource state *X* at time *t* and information regarding the relative degree of credibility in each model at time *t*, a single best decision is made that drives the system to a new state at time *t*+1. Following the decision, the system is observed at time *t*+1, and the observation is compared to predictions of system state, X^*, generated by each alternative model. Through application of Bayes' Rule, these comparisons update the information state and thus the degree of credibility allocated to each model at time *t*+1, at which time the process is repeated.

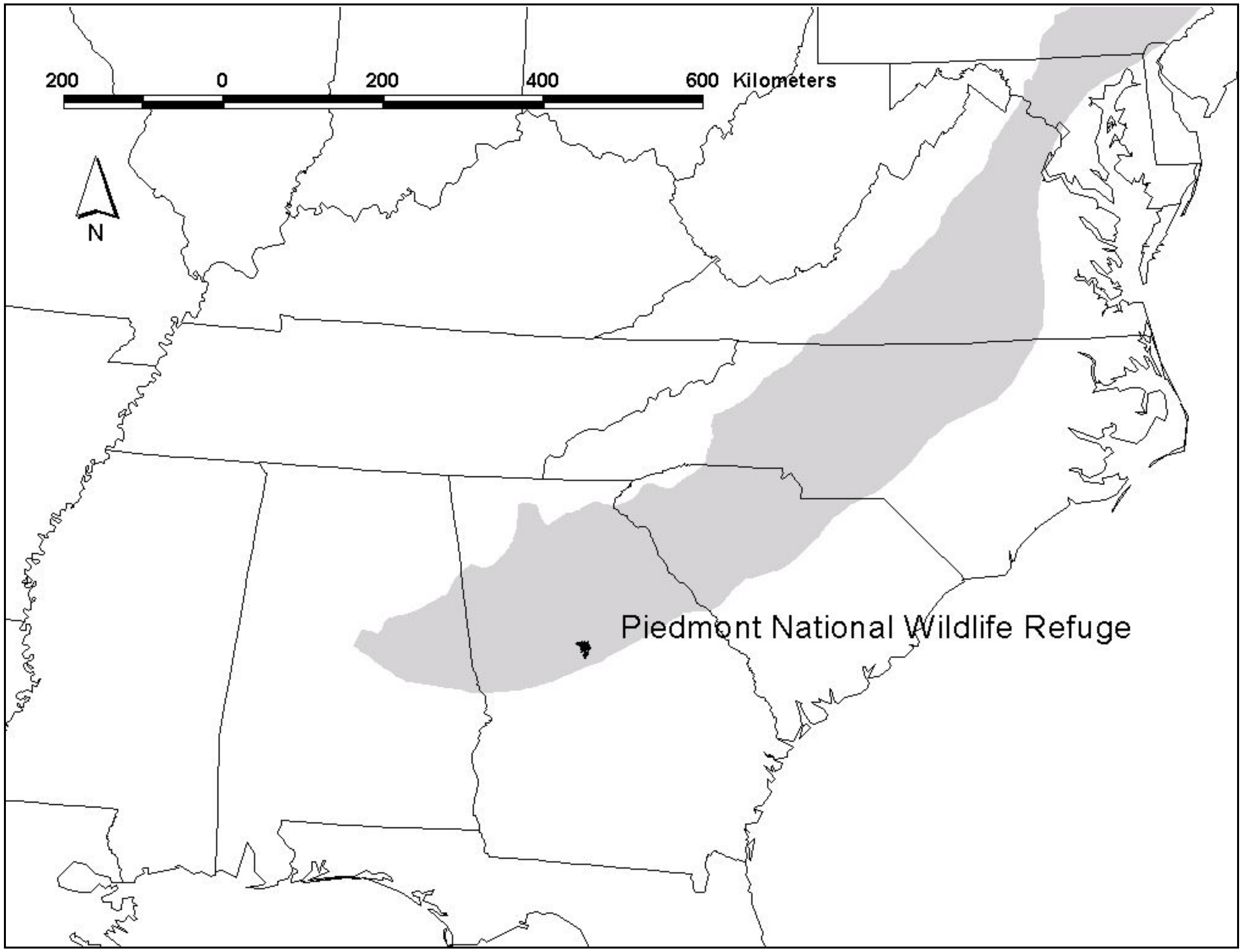


Figure 5. Piedmont National Wildlife Refuge, Georgia, USA, and southern extent of Piedmont physiographic province (shaded).

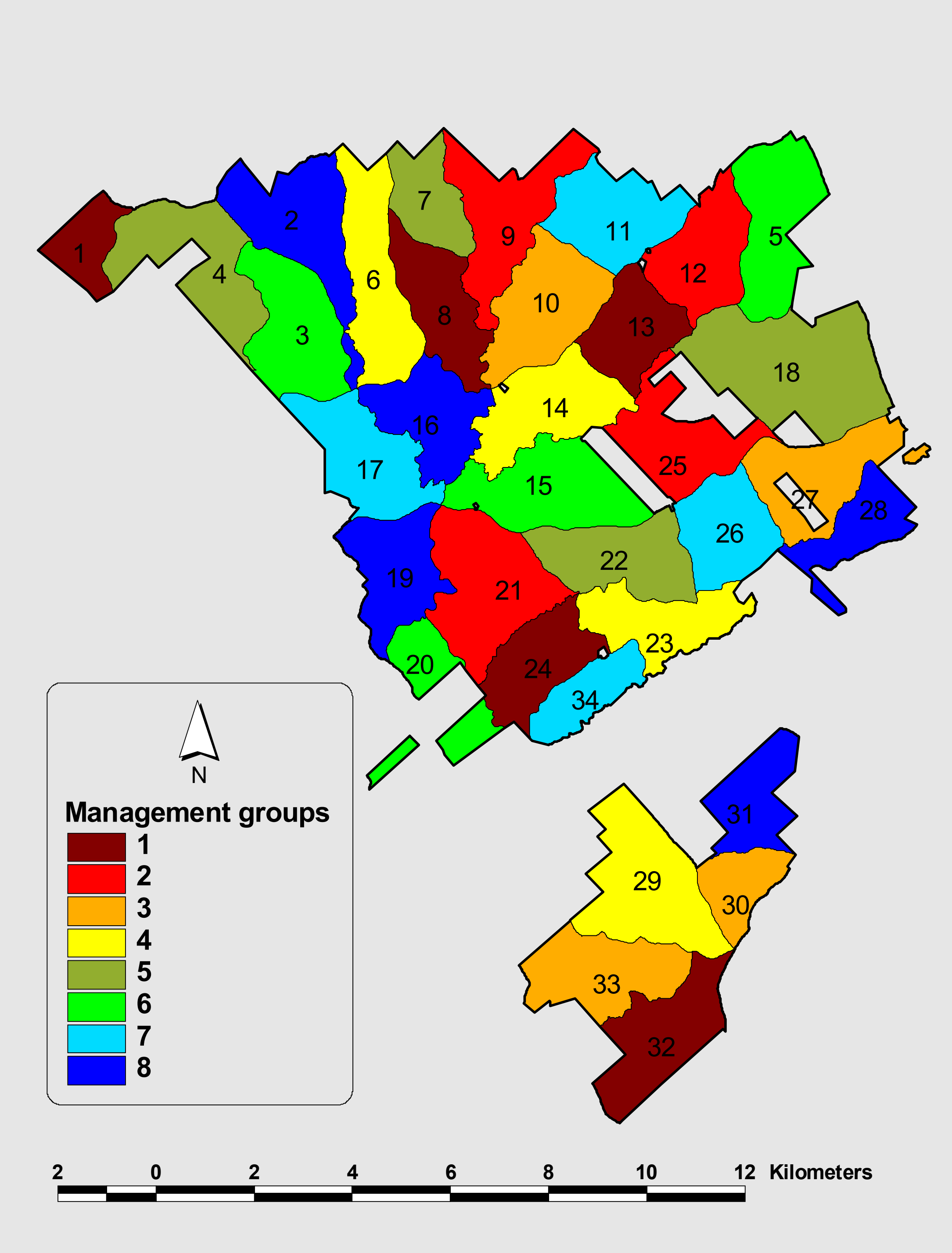


Figure 6. Management group assignments of Piedmont National Wildlife Refuge compartments.

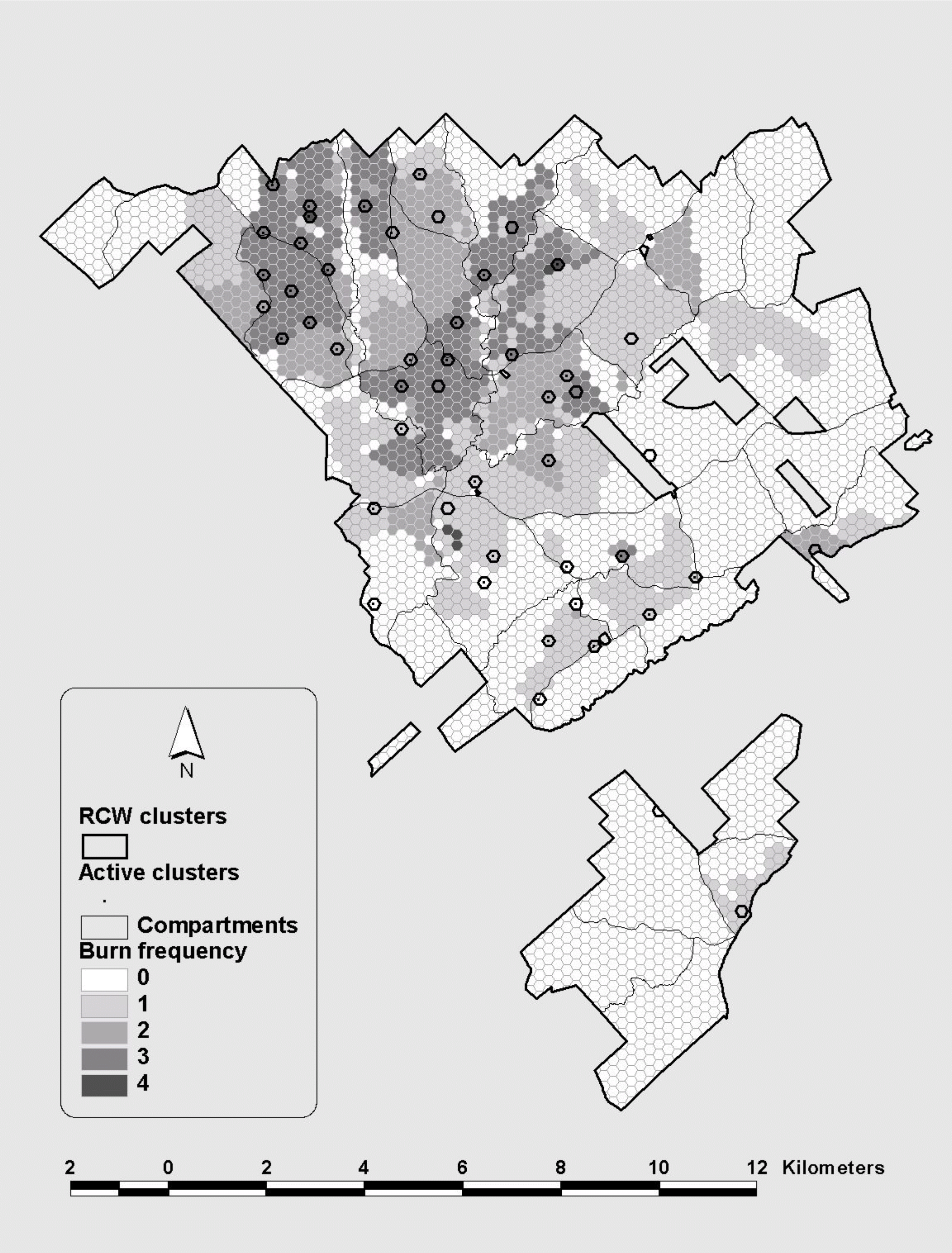


Figure 7. Burn frequency 1991-2000 for Piedmont National Wildlife Refuge. Hexagons containing a red-cockaded woodpecker cluster in year 2000 are indicated.

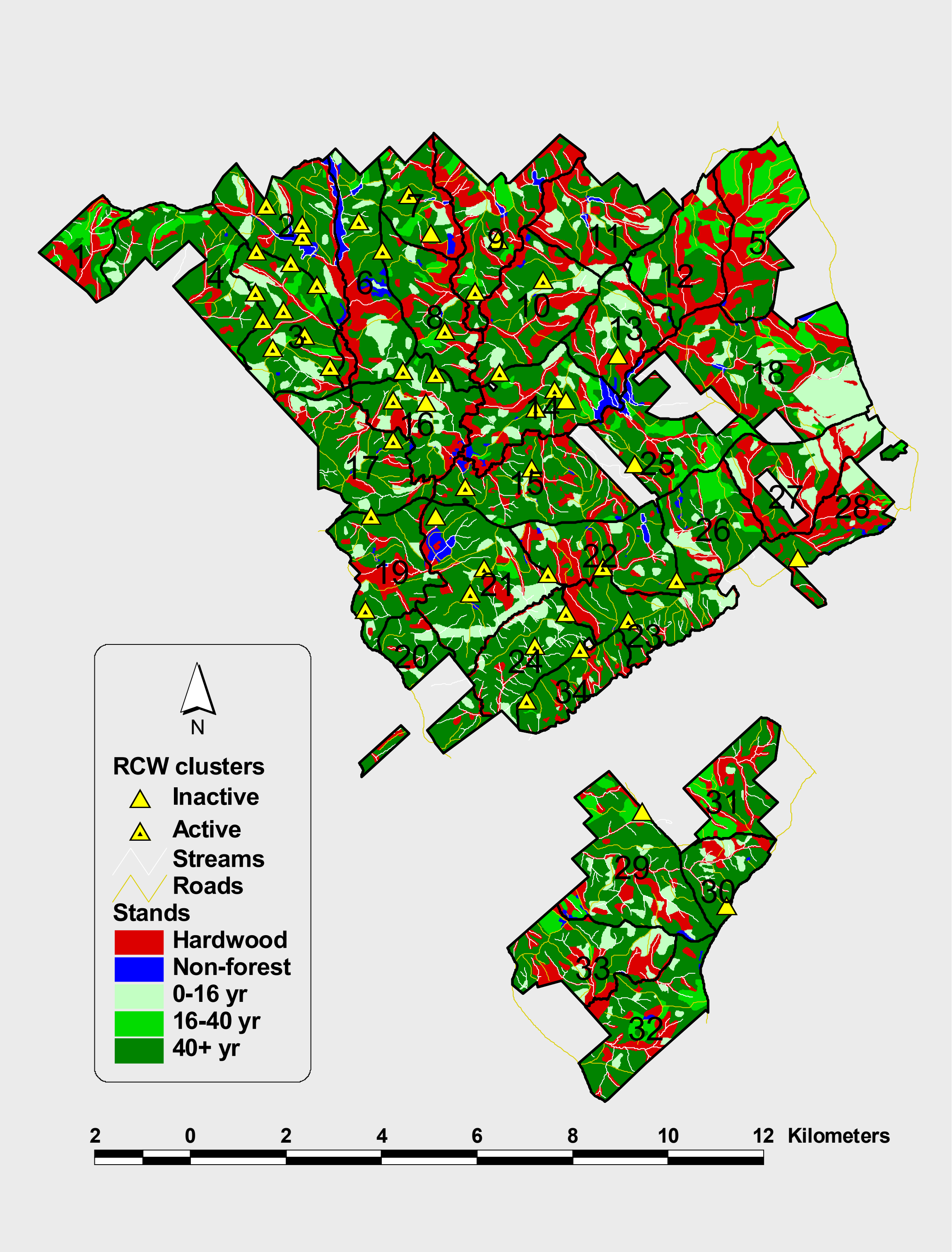


Figure 8. Forest stands, compartments, roads, streams, and red-cockaded woodpecker cluster locations displayed as GIS themes for the Piedmont National Wildlife Refuge.

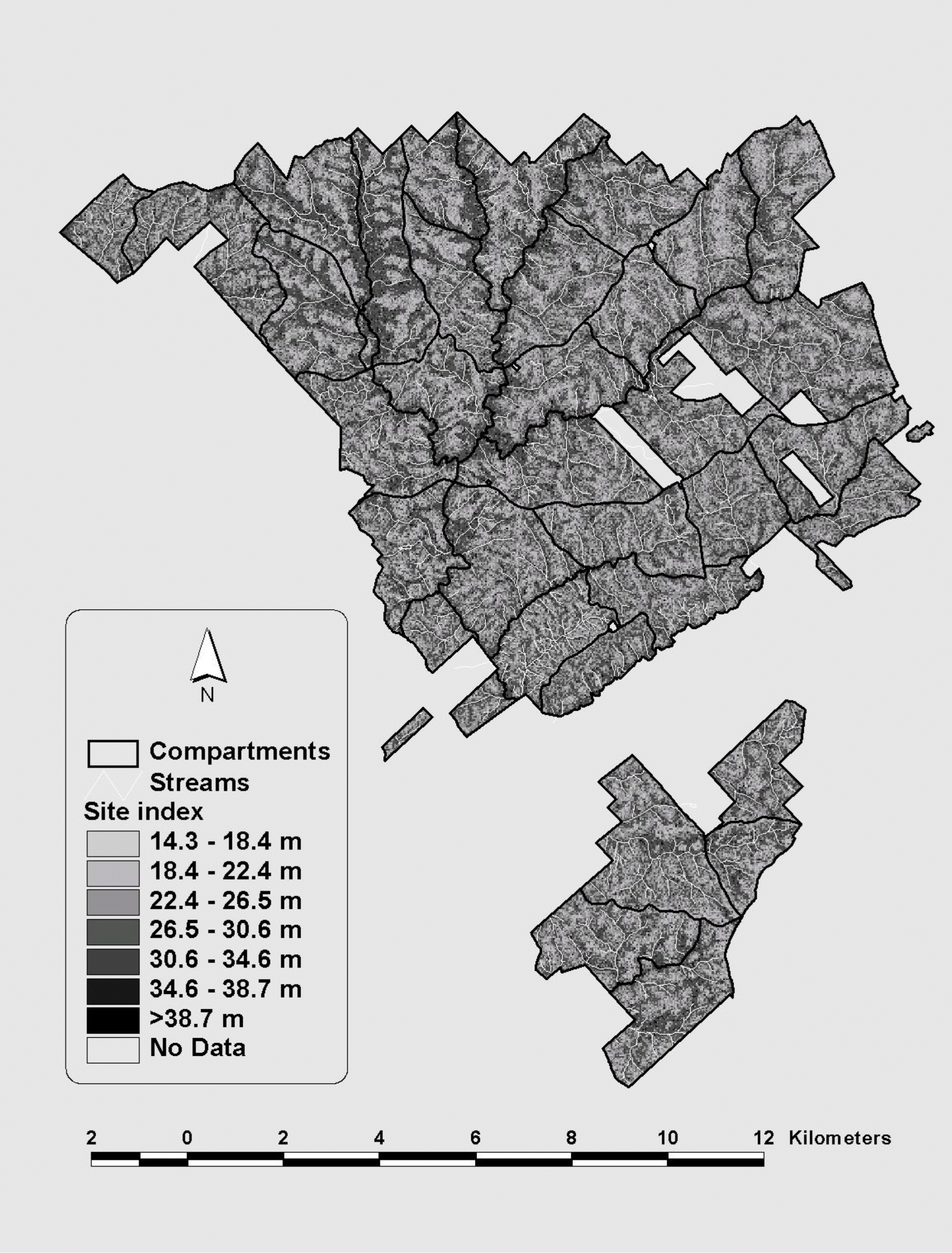


Figure 9. Predicted site index (base year 50) for Piedmont National Wildlife Refuge.

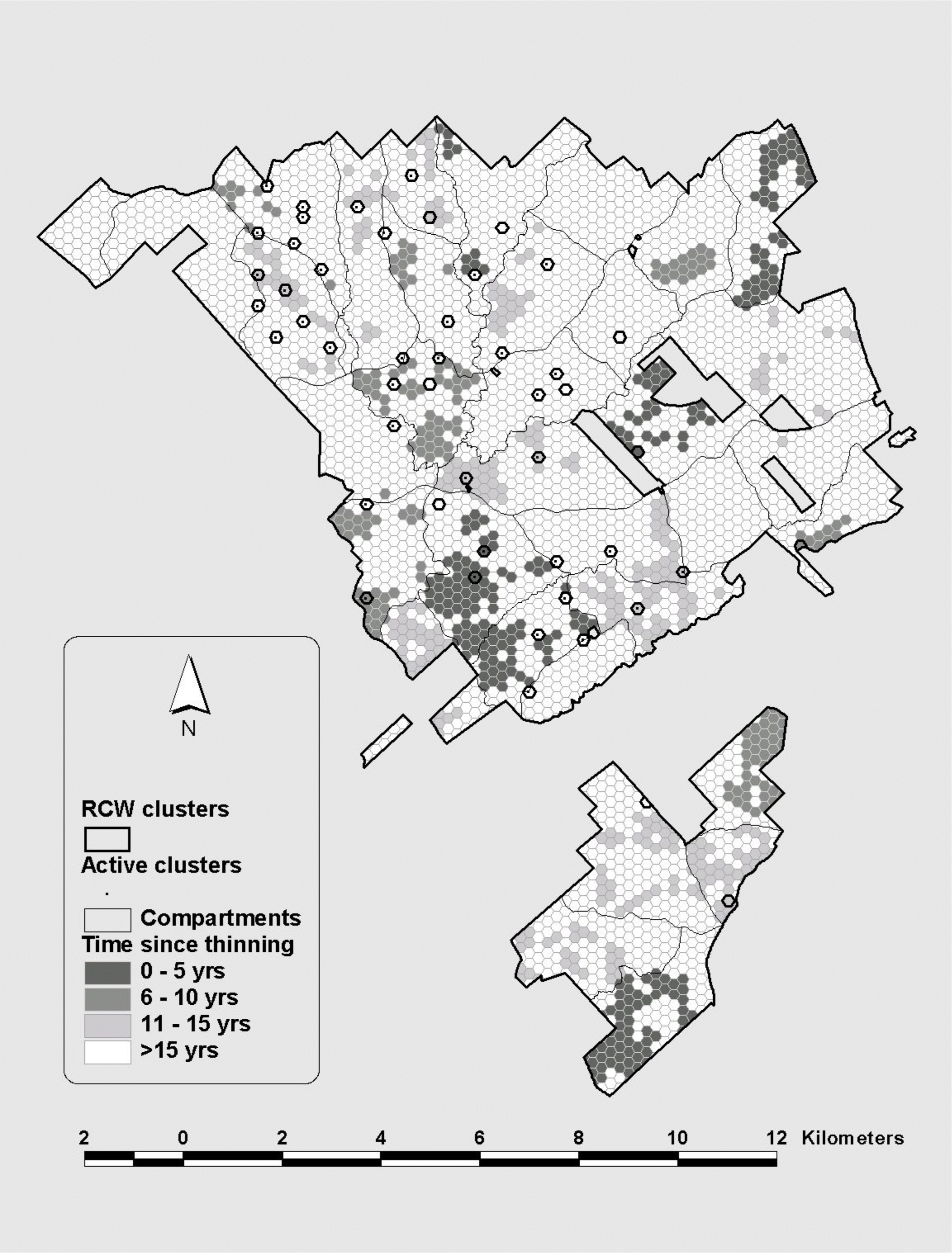


Figure 10. Time since last thinning in year 2000 for Piedmont National Wildlife Refuge. Hexagons containing a red-cockaded woodpecker cluster are indicated.

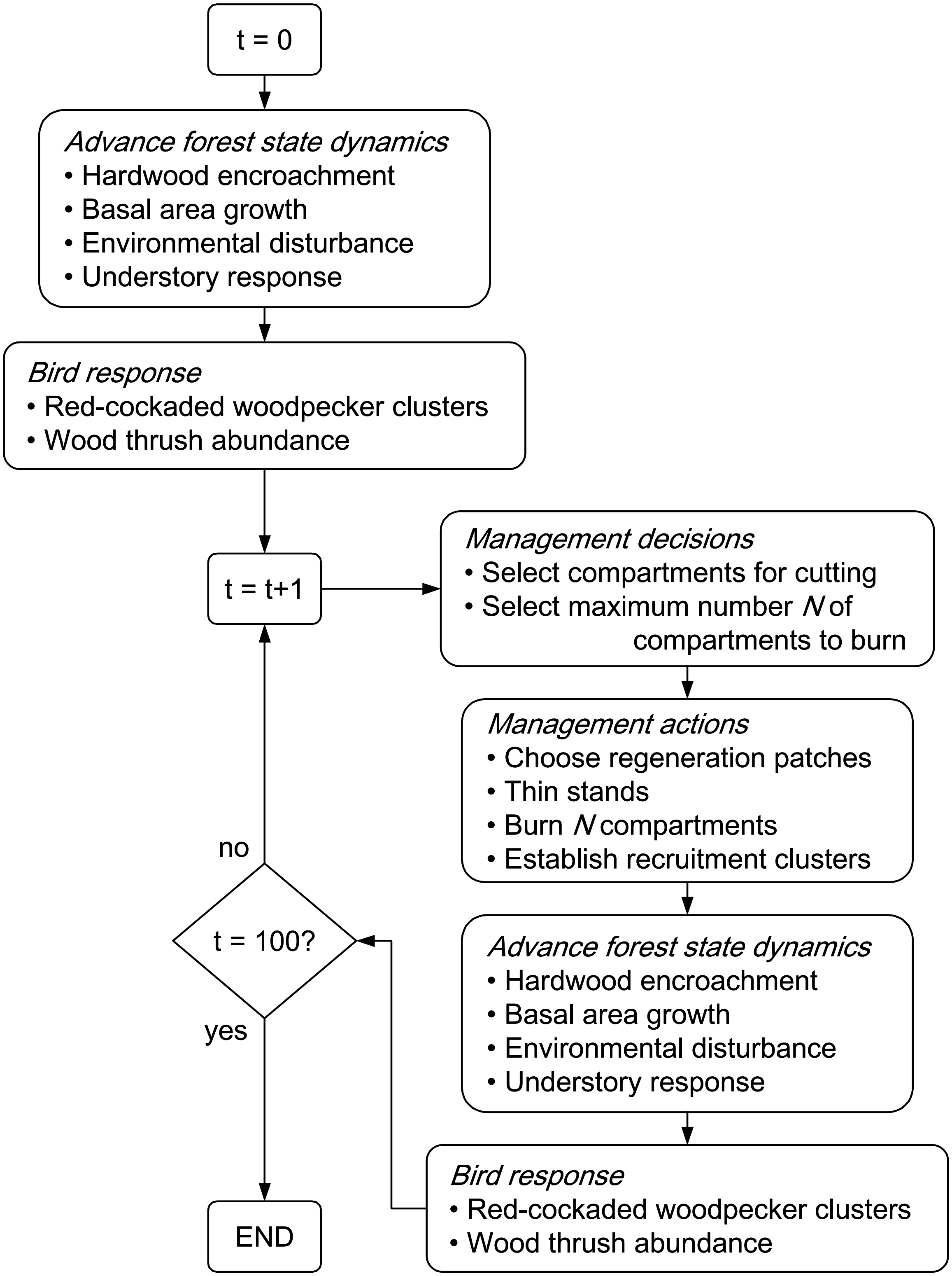
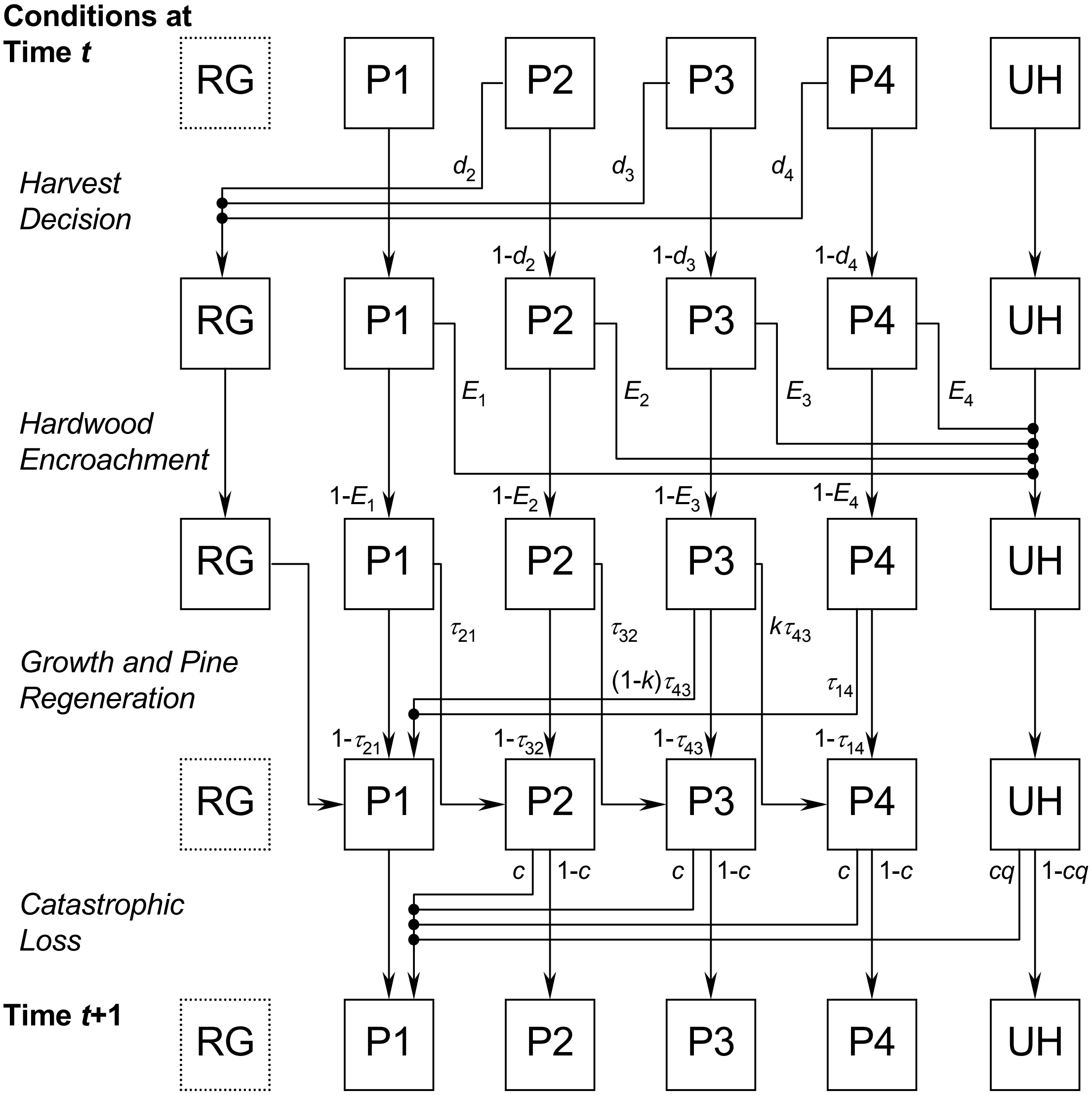


Figure 11. Sequence of management decisions, management actions, forest responses, and bird population responses portrayed in Refuge management simulation model.

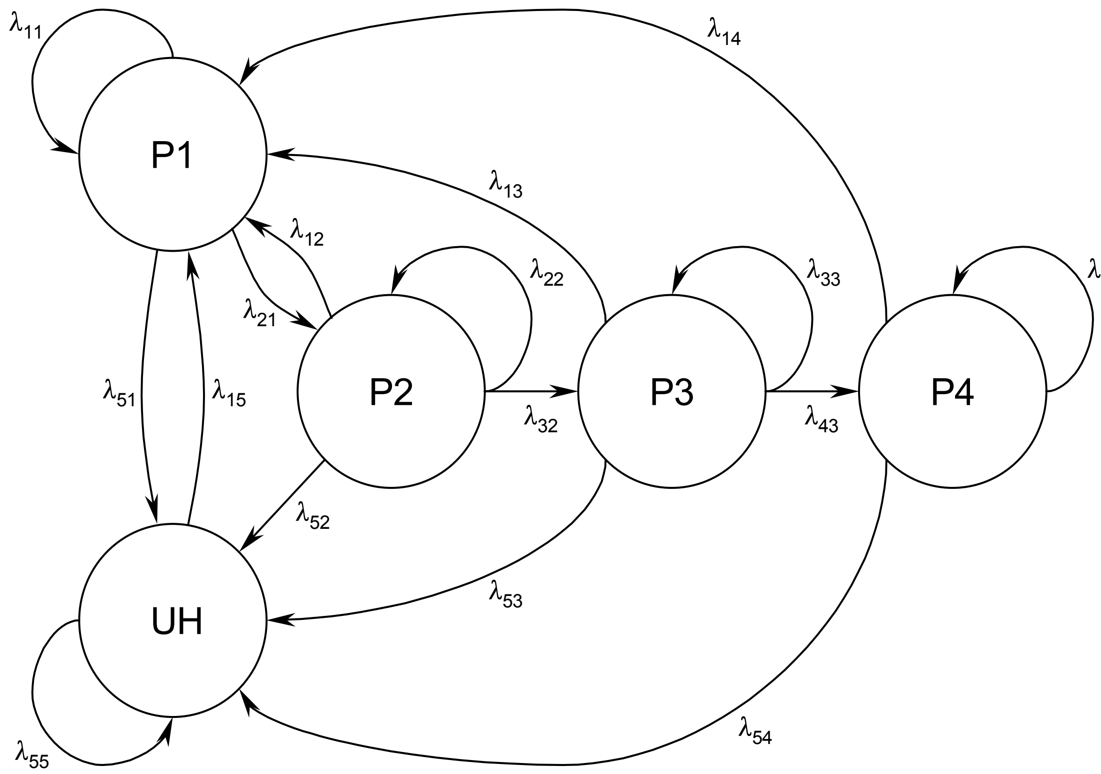


$$E_i = \max(0, e_i - d_i)$$

Model Parameters		
Decisions	d_2, d_3, d_4	
Encroachment rates*	e_1, e_2, e_3, e_4	(0.0006, 0.015, 0.015, 0.03)
Age class transitions	$\tau_{21}, \tau_{32}, \tau_{43}, \tau_{14}$	(1/16, 1/24, 1/40, 1/40)
P4 admission rate	k	(0.5)
Catastrophic loss rate*	c	(0.003394)
UH → P1 conversion rate	q	(0.25)

*stochastic parameters

Figure 12. Transitions among cover types of forest overstory model, displayed for successive processes within a single time step.



$\lambda_{11} = (1\text{-}\tau_{21}+c\tau_{21})(1\text{-}E_1)+cqE_1$
$\lambda_{12} = d_2+(c(1\text{-}E_2)+cqE_2)(1\text{-}d_2)$
$\lambda_{13} = d_3+(((1\text{-}k)\tau_{43}+c(1\text{-}\tau_{43})+ck\tau_{43})(1\text{-}E_3)+cqE_3)(1\text{-}d_3)$
$\lambda_{14} = d_4+((\tau_{14}+c(1\text{-}\tau_{14}))(1\text{-}E_4)+cqE_4)(1\text{-}d_4)$
$\lambda_{15} = cq$
$\lambda_{21} = (1\text{-}c)\tau_{21}(1\text{-}E_1)$
$\lambda_{22} = (1\text{-}c)(1\text{-}\tau_{32})(1\text{-}E_2)(1\text{-}d_2)$
$\lambda_{32} = (1\text{-}c)\tau_{32}(1\text{-}E_2)(1\text{-}d_2)$
$\lambda_{33} = (1\text{-}c)(1\text{-}\tau_{43})(1\text{-}E_3)(1\text{-}d_3)$
$\lambda_{43} = (1\text{-}c)k\tau_{43}(1\text{-}E_3)(1\text{-}d_3)$
$\lambda_{44} = (1\text{-}c)(1\text{-}\tau_{14})(1\text{-}E_4)(1\text{-}d_4)$
$\lambda_{51} = (1\text{-}cq)E_1$
$\lambda_{52} = (1\text{-}cq)E_2(1\text{-}d_2)$
$\lambda_{53} = (1\text{-}cq)E_3(1\text{-}d_3)$
$\lambda_{54} = (1\text{-}cq)E_4(1\text{-}d_4)$
$\lambda_{55} = 1\text{-}cq$

Figure 13. Total single-step transition probabilities between forest cover types in forest overstory model.

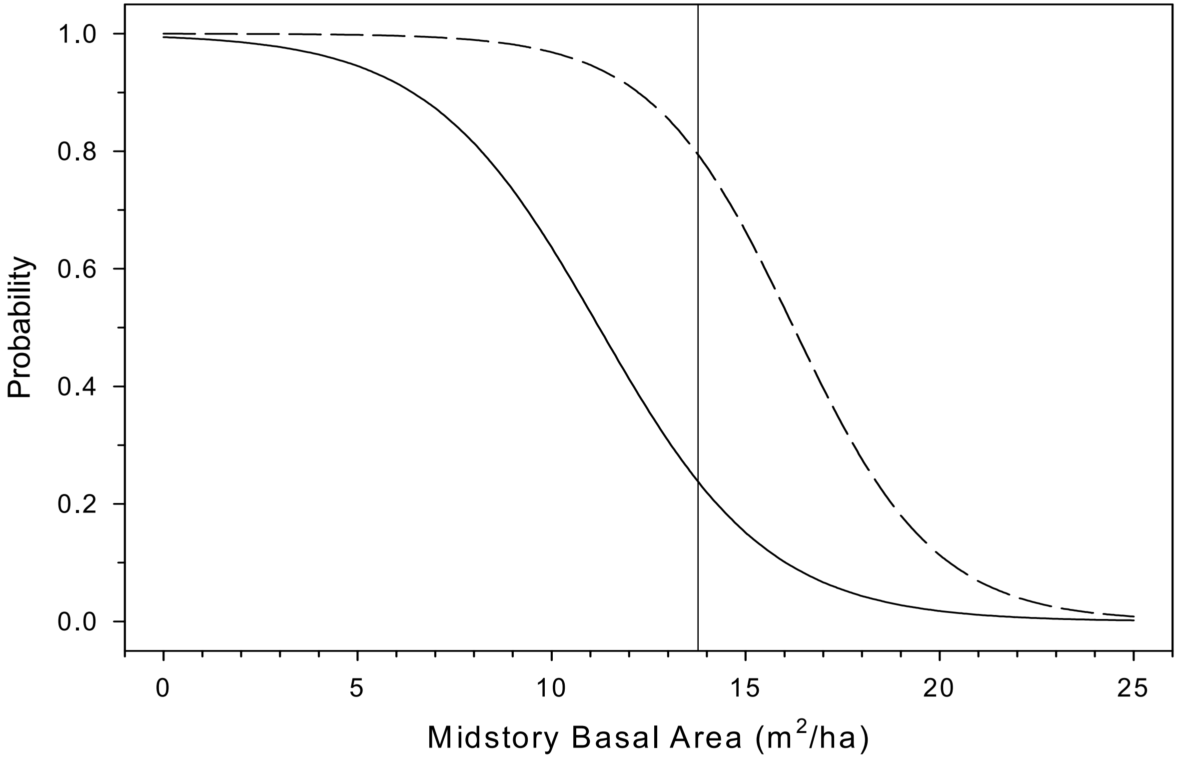


Figure 14. Model of Loeb et al. (1992) for unconditional probability of red-cockaded woodpecker cluster occupancy as a function of midstory basal area (solid line), and an alternative model representing probability of persistence conditional on cluster occupation (dashed line). Models were most distinct near 13.8 m^2/ha (60 ft^2/ac, vertical reference line), where approximate unconditional and conditional probabilities were 0.20 and 0.80, respectively.

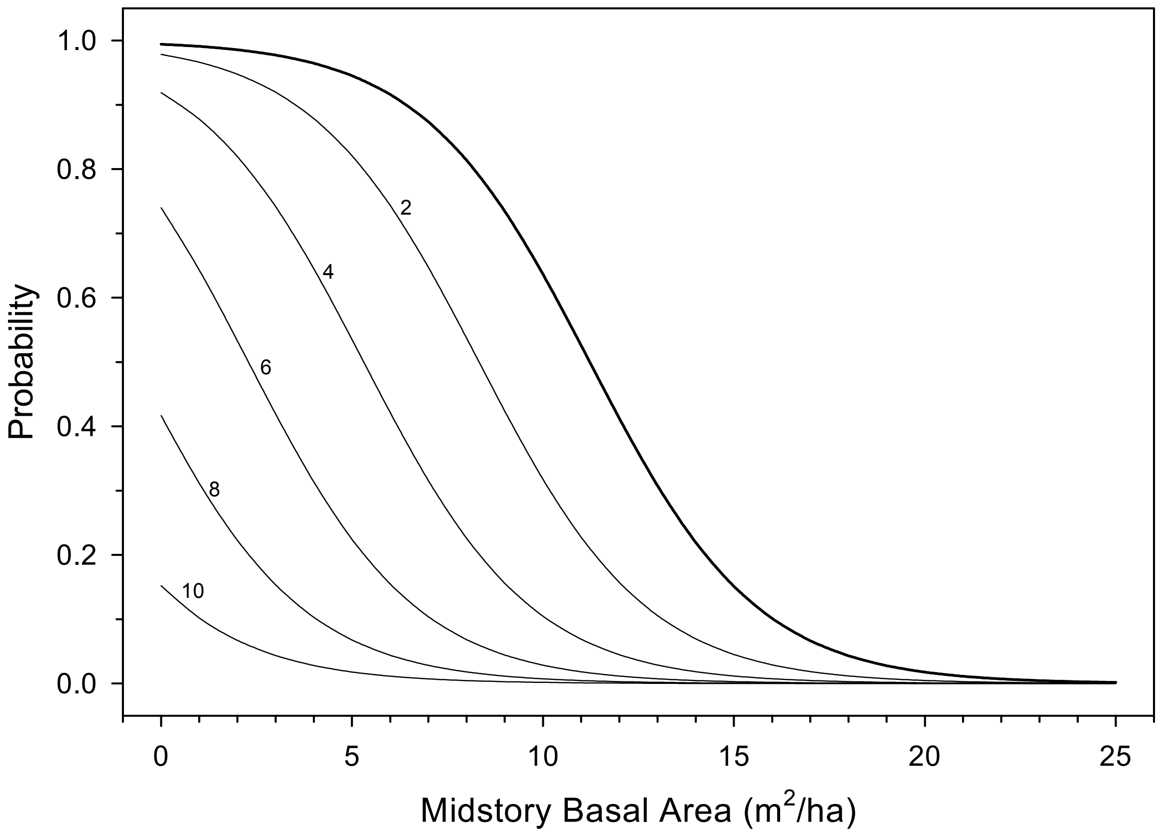


Figure 15. Model of Loeb et al. (1992) for unconditional probability of red-cockaded woodpecker cluster occupancy as a function of midstory basal area (heavy solid line), and an alternative model reflecting probability of inactive cluster settlement displayed at different levels of cluster reproductive isolation.

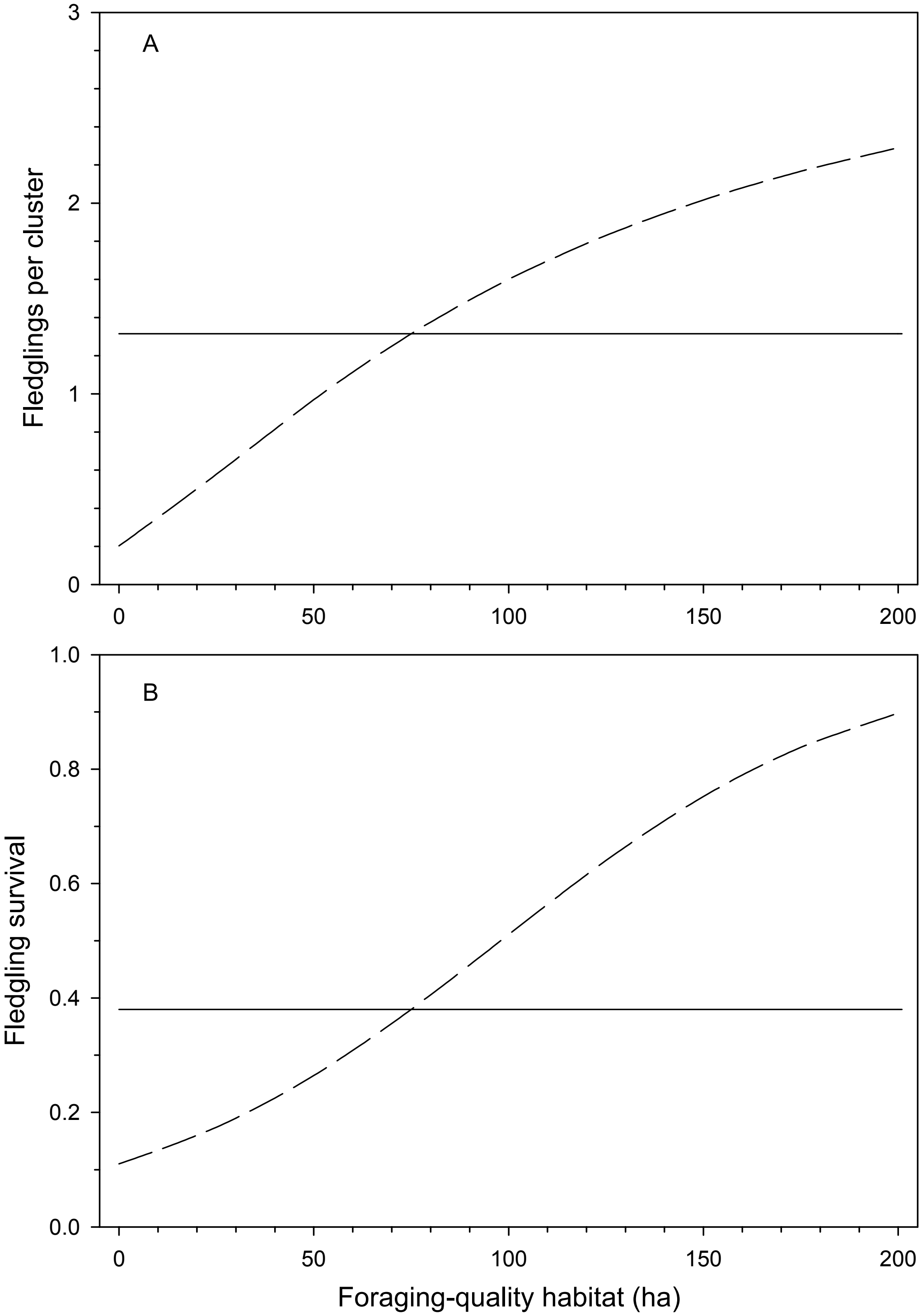


Figure 16. Alternative models of red-cockaded woodpecker fledglings per active cluster (A) and fledgling survival (B). Models proposed that responses were either insensitive (solid line) or sensitive (dashed line) to amount of foraging-quality habitat.

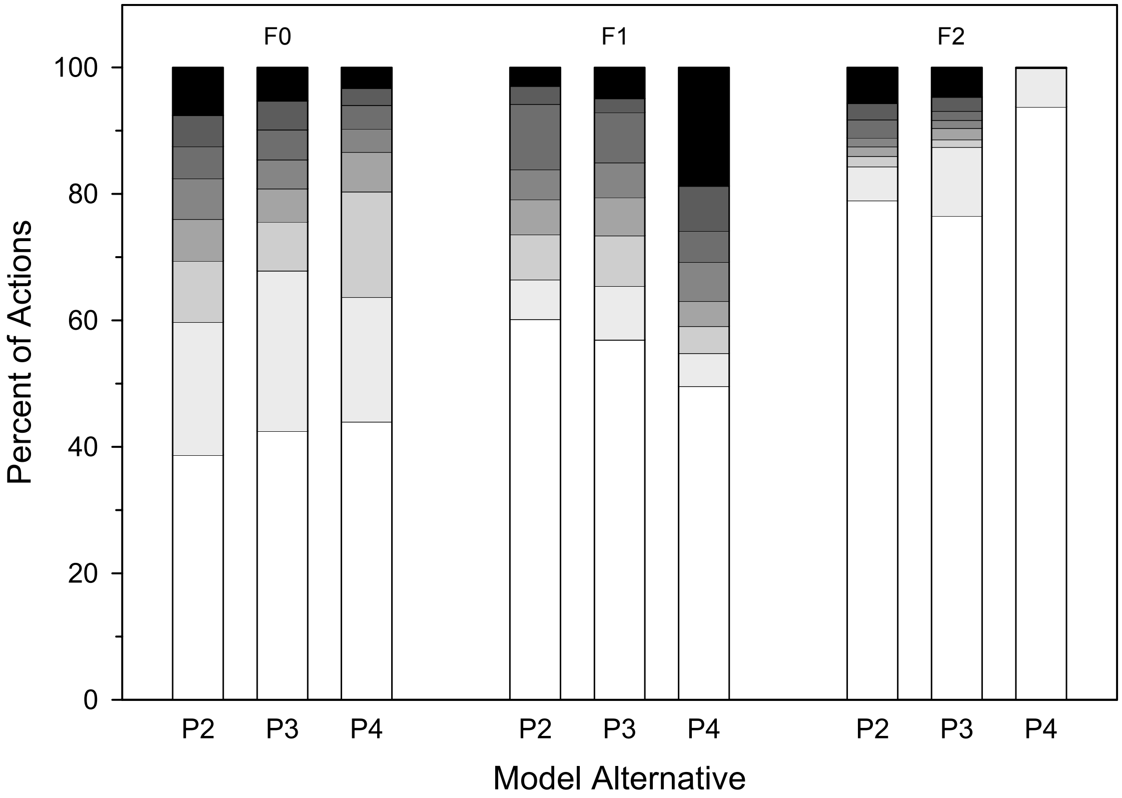


Figure 17. Sensitivity of optimal decision policy to parameterization of overstory transition model. Distributions of three types of decision actions, amount of regeneration cutting in P2, P3, and P4 stand classes, are displayed for each of the three model alternatives: original model (F0), model of low transition into P4 (F1), and model of high transmission into P4 (F2). The eight decision actions, a series of regeneration quantities 0.00, 0.02, ..., 0.14, are expressed in order by progressive light-to-dark shading.

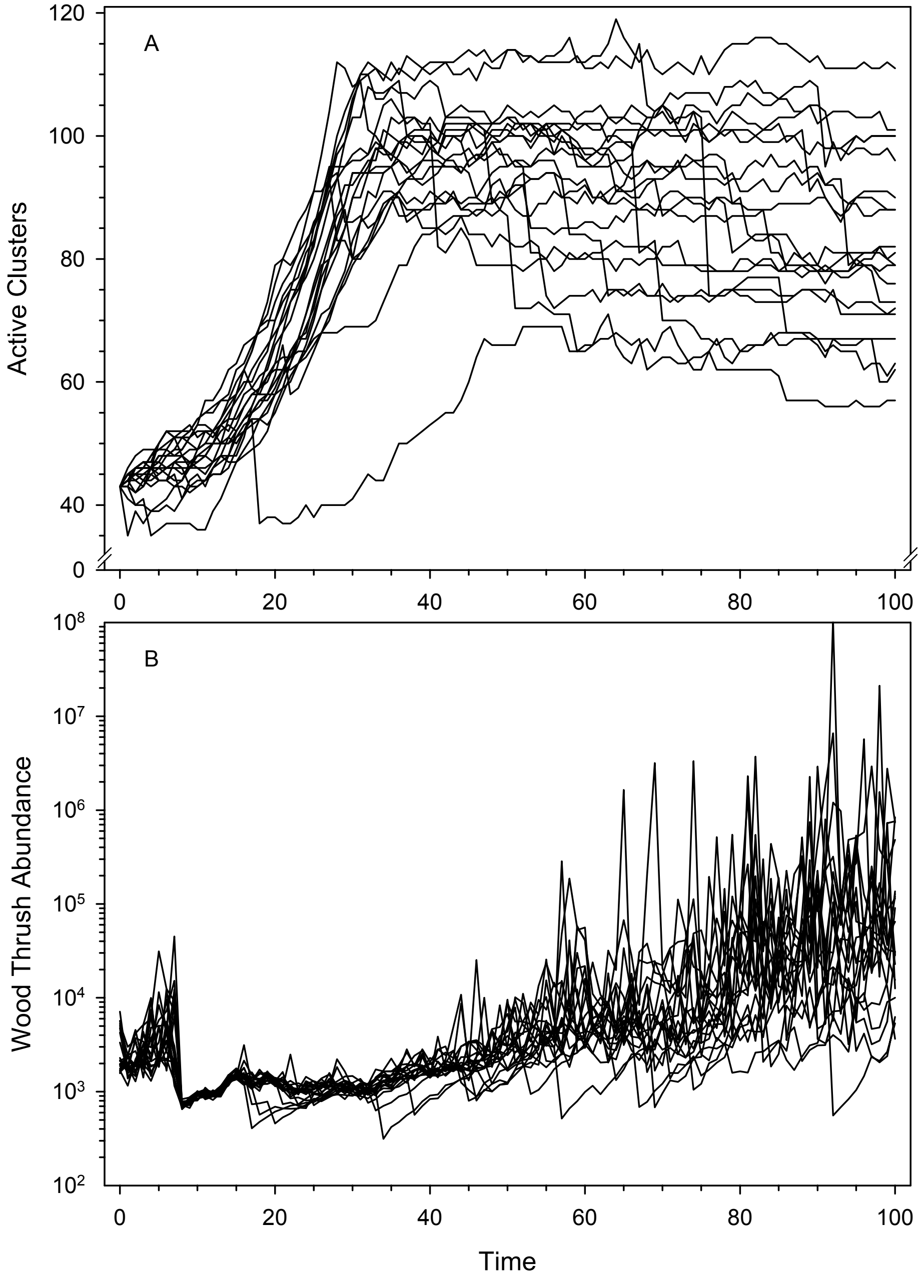


Figure 18. Red-cockaded woodpecker (A) and wood thrush (B) population responses in twenty replicate runs of the spatially-explicit forest management simulation model. Results are displayed for model M_{000} in which overstory transition into the old-growth class (P4) was moderate, woodpecker productivity was insensitive to foraging habitat, and wood thrush density was linearly related to habitat measures. Management decisions are simulated under the current Refuge compartment permutation sequence and under a low regime of burning (seven compartments burned annually).

APPENDIX C

PROGRAM LISTINGS

Appendix C.1. *Read Grid Data.SAS*. SAS program reads and merges DEM grid-based ASCII data files exported from ArcView. Files merged are slope, aspect, curvature, slope position, and the DEM. Output is saved as the SAS file PNWRGRID.

```
** READ GRID DATA.SAS **;

** Libname for permanent SAS data file of grid information **;
libname out 'c:\giscov\dem2';

** Location of ASCII export file of elevation values **;
filename in1 'c:\giscov\dem2\refuge_dem.asc';

** Location of ASCII export file of slope values **;
filename in2 'c:\giscov\dem2\refuge_slope.asc';

** Location of ASCII export file of aspect values **;
filename in3 'c:\giscov\dem2\refuge_aspect.asc';

** Location of ASCII export file of curvature values **;
filename in4 'c:\giscov\dem2\refuge_curvature.asc';

** Location of ASCII export file of slope position values **;
filename in5 'c:\giscov\dem2\refuge_slope_pos.asc';

** Read elevation grid data.  Limit for COL variable must be the
   same as NCOLS= value in header of ASCII file **;
data a1;
  infile in1 lrecl=10000;
  if _n_=1 then input /////;
  row = _n_;
  do col=1 to 1179;
    input elev @;
    output;
  end;
  run;

proc univariate data=a1;  var row col elev;  run;

** Delete missing values **;
data a1;
  set a1;
  if elev^=-9999;
  run;

proc univariate data=a1;  var row col elev;  run;
```

```
** Read slope grid data.  Limit for COL variable must be the
   same as NCOLS= value in header of ASCII file **;
data a2;
  infile in2 lrecl=10000;
  if _n_=1 then input /////;
  row = _n_;
  do col=1 to 1179;
    input slope @;
    output;
  end;
  run;

proc univariate data=a2;  var row col slope;  run;

** Delete missing values **;
data a2;
  set a2;
  if slope^=-9999;
  run;

proc univariate data=a2;  var row col slope;  run;

** Read aspect grid data.  Limit for COL variable must be the
   same as NCOLS= value in header of ASCII file **;
** Read grid data;
data a3;
  infile in3 lrecl=10000;
  if _n_=1 then input /////;
  row = _n_;
  do col=1 to 1179;
    input aspect @;
    output;
  end;
  run;

proc univariate data=a3;  var row col aspect;  run;

** Delete missing values **;
data a3;
  set a3;
  if aspect^=-9999;
```

Appendix C.1. *Read Grid Data.SAS* Continued.

```
  run;

proc univariate data=a3;  var row col aspect;  run;

** Read curvature grid data.  Limit for COL variable must be the
   same as NCOLS= value in header of ASCII file **;
data a4;
  infile in4 lrecl=10000;
  if _n_=1 then input /////;
  row = _n_;
  do col=1 to 1179;
    input curvatur @;
    output;
  end;
  run;

proc univariate data=a4;  var row col curvatur;  run;

** Delete missing values **;
data a4;
  set a4;
  if curvatur^=-9999;
  run;

proc univariate data=a4;  var row col curvatur;  run;

** Read slope position grid data.  Limit for COL variable must be the
   same as NCOLS= value in header of ASCII file **;
data a5;
  infile in5 lrecl=10000;
  if _n_=1 then input /////;
  row = _n_;
  do col=1 to 1179;
    input slopepos @;
    output;
  end;
  run;

proc univariate data=a5;  var row col slopepos;  run;

** Delete missing values **;
data a5;
  set a5;
  if slopepos^=-9999;
  run;

proc univariate data=a5;  var row col slopepos;  run;

** Create permanent dataset by combining A1-A5 **;
data out.pnwrgrid;
  merge a1 a2 a3 a4 a5;
  by row col;
  run;

** Univariate and frequency summaries of combined dataset **;
proc univariate data=out.pnwrgrid;
  var elev slope aspect curvatur slopepos;
  run;

proc freq data=out.pnwrgrid;
  table slopepos;
  run;
```

Appendix C.2. *Analyze Grid Data.SAS*. Given a set of site index model parameters, SAS program estimates site index for every 30-m exported grid cell and computes summaries of the index value for 1% groups of cells selected at random. Cell values of site index are exported to an ASCII file for later import into ArcView.

```
** ANALYZE GRID DATA.SAS **;

** Input site index prediction model parameters here **;
%let b0 = log(99/3.280839895);
%let b1 = 0.08;
%let b2 = 0.08;
%let b3 = -0.13;
%let b4 = -0.074;
%let b5 = 0;
%let s = 0.103;

** Set a seed value for random number generator (0 = clock-set) **;
%let ranseed = 868521597;

** Libname of SAS database with grid information **;
libname out 'c:\giscov\dem2';

** Location of an ASCII grid export file **;
filename in1 'c:\giscov\dem2\refuge_dem.asc';

** Location of ASCII output file of site index values **;
filename in2 'c:\giscov\dem2\refuge_si.asc';
filename in2 'c:\temp\refuge_si.asc';

** Apply site index model;
data a;
  set out.pnwrgrid;
  dev = &s*rannor(&ranseed);
  rannum = ranuni(&ranseed);
  ** GROUP is a number (1-100) that represents a 1% sampling group
        of all pixels in dataset **;
  group = ceil(rannum*100);
  rads = aspect*arcos(0)/180;
  slope = slope/100;
  pred = &b0 + &b1*slope*cos(rads) + &b2*slope*sin(rads) + &b3*slope
             + &b4*(slopepos-1) + &b5*(slopepos-1)**2 + dev;
  si = exp(pred)*3.280839895;
  if slopepos=1 then bottom = 1;
  else bottom = 0;
  run;

proc univariate data=a;
  var si;
  run;

** Statistics for site index by GROUP number **;
title1 "Parameter set (&b0, &b1, &b2, &b3, &b4, &b5, &s)";
proc means noprint data=a n mean std min max;
  class group bottom;
  var si;
  output out=b n=n_si mean=mean_si std=std_si min=min_si max=max_si;
  run;

proc sort data=b (where=(_type_=3)) out=b1;  by bottom group;  run;

proc print data=b1;
  by bottom;
  var group n_si mean_si std_si min_si max_si;
  run;

** Area-wide (BOTTOM=.) and habitat-specific site index statistics **;
proc print data=b (where=(_type_<=1));
  var bottom n_si mean_si std_si min_si max_si;
  run;

** Means of GROUP-based statistics for site index **;
proc means data=b (where=(_type_=3)) mean nway;
  class bottom;
  var n_si mean_si std_si min_si max_si;
  run;

** Create a template for grid data export.
   Upper limit for COL has to be the same as the NCOLS value
   in the ASCII file.  **;
data a1;
  infile in1 lrecl=10000;
  if _n_=1 then input /////;
  row = _n_;
```

Appendix C.2. *Analyze Grid Data.SAS* Continued.

```
  do col=1 to 1179;
    input elev @;
    output;
  end;
  run;

** Merge site index values data with grid data, then export grid.
   The header information that appears after the IF _N_=1 line should
   be copied from the ASCII file read above **;
data _null_;
  merge a1 a (keep=row col si);
  by row col;
  file in2 lrecl=20000;
  if si=. then si = -9999;
  if _n_=1 then put
     'ncols         1179' /
     'nrows         1406' /
     'xllcorner     231390.022207'  /
     'yllcorner     3640510.733316' /
     'cellsize      30'   /
     'NODATA_value  -9999';
  if ^last.row then put si @;
  else put si;
  run;
```

Appendix C.3. *Hexagon Grid Create.SAS*. SAS program creates a text file of vertices for creation of a hexagon grid theme in ArcInfo.

```
** Hexagon grid create.SAS ***;
** Computes coordinates of a RxC hexagon grid **;

** USER INPUT AREA *************************

  OUTDIR:  Specify name of directory where output file is to be written
  OUTFILE: Name of output file
  R:       Number of rows of hexagon grid
  C:       Number of columns of hexagon grid
  X0:      X-coordinate (longitude) of center of SW hexagon
  Y0:      Y-coordinate (latitude) of center of SW hexagon
  A:       Hexagon area (keep in same units as X0, Y0);

%let outdir = c:\refuge arm\mapping;
%let x0 = 237300;
%let y0 = 3653000;

** To calculate r and c, given map width X and height Y **
**   Calculate h = sqrt(a/(2*sqrt(3)))
**   Then c = ceil(X/(sqrt(3)*h)) + 1
**        r = ceil(Y/(2*h)) + 1        **;

** 20 acres **;
%let outfile = hex20.file;
%let r = 69;
%let c = 72;
%let a = 80937.12842;      ** (80937 m^2 = 20-acre hexagons) **;

** 10 acres **;
%let outfile = hex10.file;
%let r = 98;
%let c = 101;
%let a = 40468.56421;

** test **;
%let outfile = hextest.file;
%let r = 8;
%let c = 10;
%let a = 80937.12842;      ** (80937 m^2 = 20-acre hexagons) **;

*********************************************;
```

```
filename outdat "&outdir";

data _null_;
  file outdat("&outfile") ls=140;
  r = &r;
  c = &c;
  x0 = &x0;
  y0 = &y0;
  a = &a;
  h = sqrt(a/(2*sqrt(3)));
  w = h*sqrt(3)/3;
  v = h*2*sqrt(3)/3;
  index = 0;
  x = x0;
  y = y0;
  do i=1 to r;
    do j=1 to c;
      index = index+1;
      x = x0 + (j-1)*(v+w);
      y = y0 + (i-1)*2*h + (mod(j,2)=0)*h;
      x1 = x-w;  y1 = y-h;
      x2 = x+w;  y2 = y-h;
      x3 = x+v;  y3 = y;
      x4 = x+w;  y4 = y+h;
      x5 = x-w;  y5 = y+h;
      x6 = x-v;  y6 = y;
*     put @1 index 4.0 i 2.0 j 2.0 (x1 y1 x2 y2 x3 y3 x4 y4 x5 y5 x6 y6) (10.1);
      put @1 index 5.0 @6 ',' @8 x1 11.3 @19 ',' @21 y1 11.3 @32 ',' /
                              @8 x2 11.3 @19 ',' @21 y2 11.3 @32 ',' /
                              @8 x3 11.3 @19 ',' @21 y3 11.3 @32 ',' /
                              @8 x4 11.3 @19 ',' @21 y4 11.3 @32 ',' /
                              @8 x5 11.3 @19 ',' @21 y5 11.3 @32 ',' /
                              @8 x6 11.3 @19 ',' @21 y6 11.3 @32 ',' /
                              @8 x1 11.3 @19 ',' @21 y1 11.3 @32 ',' / ' END';
    end;
  end;
  put 'END';
  run;
```

Appendix C.4. *Hexagon ID Assign.SAS*. SAS program reads the database for the hexagon grid theme, and it adds a new identifier field and coordinates of hexagon centers.

```
** Hexagon ID assign.SAS ***;
** Reads a hexagon attribute file from ArcView, attaches a new identifier,
    hexagon row and column indices, and hexagon center point coordinates   **;


** USER INPUT AREA *************************

  OUTDIR:  Specify name of directory where output file is to be written
  OUTFILE: Name of output file (DBF format)
  INFILE:  Name of input file (DBF format)
  DBF_ID:  Name of hexagon ID variable in input file
  R:       Number of rows of hexagon grid
  C:       Number of columns of hexagon grid
  X0:      X-coordinate (longitude) of center of SW hexagon
  Y0:      Y-coordinate (latitude) of center of SW hexagon
  A:       Hexagon area (keep in same units as X0, Y0);


** To calculate r and c, given map width X and height Y **
**   Calculate h = sqrt(a/(2*sqrt(3)))
**   Then c = ceil(X/(sqrt(3)*h)) + 1
**        r = ceil(Y/(2*h)) + 1       **;

/***
** 20 acres **;
%let r = 69;
%let c = 72;
%let a = 80937.12842;      ** (80937 m^2 = 20-acre hexagons) **;
%let x0 = 237300;
%let y0 = 3653000;
%let outdir = c:\refuge arm\mapping;
%let infile = hex20 (original).dbf;
%let outfile = hex20.dbf;
%let dbf_id = hex2_;
***/

** 10 acres **;
%let r = 98;
%let c = 101;
%let a = 40468.56421;      ** (40469 m^2 = 10-acre hexagons) **;
%let x0 = 237300;
%let y0 = 3653000;
%let outdir = c:\refuge arm\mapping;
%let infile = hex10 (original).dbf;
%let outfile = hex10.dbf;
%let dbf_id = hex3_;

/***
** test **;
%let r = 8;
%let c = 10;
%let a = 80937.12842;      ** (80937 m^2 = 20-acre hexagons) **;
%let x0 = 237300;
%let y0 = 3653000;
%let outdir = c:\refuge arm\mapping;
%let infile = hextest_bak.dbf;
%let outfile = hextest.dbf;
%let dbf_id = hex00_;
***/

*********************************************;


libname out "&outdir";
filename indbf "&outdir\&infile";
filename outdbf "&outdir\&outfile";

data a (keep = hex_id row col x y);
  r = &r;
  c = &c;
  x0 = &x0;
  y0 = &y0;
  a = &a;
  h = sqrt(a/(2*sqrt(3)));
  w = h*sqrt(3)/3;
  v = h*2*sqrt(3)/3;
  hex_id = 0;
  x = x0;
  y = y0;
  do row=1 to r;
    do col=1 to c;
      hex_id = hex_id+1;
```

Appendix C.4. *Hexagon ID Assign.SAS*. Continued.

```
      x = x0 + (col-1)*(v+w);
      y = y0 + (row-1)*2*h + (mod(col,2)=0)*h;
      output;
    end;
  end;
  run;

proc dbf db4=indbf out=b;  run;

data b (keep=area perimeter &dbf_id hex_id sortord);
  ** Mapping of original hexagon ID variable into new HEX_ID variable **
  ** Renumbers hexagons from SW corner to NE **;
  set b;
  ** Preserve the original record order of the input file **;
  sortord = _n_;
  r = &r;
  c = &c;
  even = (mod(c,2)=0);
  k = &dbf_id-1;
  l = mod(k,c);
  l = l + c*(l=0);
  l2 = 2*l;
  m1 = l2*(l2<=c);
  m2 = (l2-c-even)*(l2>c);
  n = m1+m2;
  p = floor((k-0.5)/c);
  q = r-1-p;
  hex_id = q*c+n;
  run;

proc sort data=b;  by hex_id;  run;

data a;
  merge b a;
  by hex_id;
  format x y 11.3;
  run;

proc print data=a;
  run;

proc sort data=a out=a (drop=sortord);  by sortord;  run;
```

```
proc dbf db4=outdbf data=a;  run;
```

Appendix C.5. *Hexagon Attributes.SAS*. SAS program merges several map themes, each intersected with the hexagon shape file, to create the SAS database of PNWR forest attributes (PNWR_HEX) used in simulation modeling. This merged file is also saved as the ArcView theme HEX10_HEXES.DBF.

```
** Hexagon Attributes.SAS ***;
** Creates SAS database file of hexagon-based habitat characteristics:
   (1) Assigns each hexagon to a PNWR compartment, based on majority-area rule.
   (2) Calculate total area, and predominant age and BA of each forest type
        within each hexagon.
   (3) Make hexagon areas consistent, and allocate pine habitat measures into
        two area classes of predominating and secondary cover types.
   (4) Merge hexagon file with hexagon coverages of (a) management history,
        (b) site index, (c) slope, (d) slope position, (e) stream length,
        (f) distances to streams and property boundaries, (g) land ownership
        status in 400, 800, 1600, and 3200-m buffers around hex centroids,
        and (h) year 2000 RCW population status.
   (5) Create permanent SAS database and print some tables.
***;


%let hexsize = 10;

*** Hexagons in ArcView are generally not of the target hexagon size.  Some
      hexagons will be smaller than the target size and others larger, but
      the weighted average of these areas equals the target size. ***;
*** These are areas of 10-acre interior hexagons created by ArcView
***   (10 acre = 40468.56421 m^2)  **;
%let int10 = area in (40435.875 40437.562 40437.563 40439.250 40482.676
                      40484.365 40486.055);
%let min10 = 40435.875;
%let max10 = 40486.055;
%let target10 = 40468.564;

*** These are areas of 20-acre interior hexagons created by ArcView
***   (20 acre = 80937.12842 m^2)  **;
%let int20 = area in (80881.125 80883.512 80885.898 80947.312 80947.313
                      80949.701 80952.090);
%let min20 = 80881.125;
%let max20 = 80952.090;
%let target20 = 80937.128;

%let int = int&hexsize;
%let minint = min&hexsize;
%let maxint = max&hexsize;
%let avgint = target&hexsize;

** USER INPUT AREA *************************

  OUTDIR:  Specify name of directory where output file is to be written
  INFILE1: Name of DBF file containing the intersected STAND and hexagon info
  INFILE2: Name of DBF file containing the intersected COMPARTMENT and hex info
  INFILE3: Name of DBF file containing the hexagon shapes
  INFILE4: Name of DBF file containing hexagon cell treatments
  INFILE5: Name of DBF file containing hexagon-based site index (ft) values
  INFILE6: Name of DBF file containing hexagon-based slope values
  INFILE7: Name of DBF file containing hexagon-based slope position values
  INFILE8: Name of DBF file containing hexagon-stream intersections
  INFILE9: Name of DBF file containing distance variables
  INFILE10: Name of DBF file containing 400-m buffer-ownership intersections
  INFILE11: Name of DBF file containing 800-m buffer-ownership intersections
  INFILE12: Name of DBF file containing 1600-m buffer-ownership intersections
  INFILE13: Name of DBF file containing 3200-m buffer-ownership intersections
  INFILE14: Name of DBF file containing year 2000 RCW cluster status
  OUTFILE: Name of output file (DBF format)
*******************************************;

%let outdir = d:\pnwrmap\hex&hexsize;
%let infile1 = hex&hexsize._stands.dbf;
%let infile2 = hex&hexsize._comps.dbf;
%let infile3 = hex&hexsize._hexes (original).dbf;
%let infile4 = hex&hexsize._fire.dbf;
%let infile5 = hex&hexsize._si.dbf;
%let infile6 = hex&hexsize._slope.dbf;
%let infile7 = hex&hexsize._position.dbf;
%let infile8 = stream_density.dbf;
%let infile9 = hex&hexsize._distances.dbf;
%let infile10 = hex&hexsize._b400_own.dbf;
%let infile11 = hex&hexsize._b800_own.dbf;
%let infile12 = hex&hexsize._b1600_own.dbf;
%let infile13 = hex&hexsize._b3200_own.dbf;
%let infile14 = rcw_hexes.dbf;
%let outfile = hex&hexsize._hexes.dbf;
```

Appendix C.5. *Hexagon Attributes.SAS*. Continued.

```
********************************************;


filename infile1 "&outdir\&infile1";
filename infile2 "&outdir\&infile2";
filename infile3 "&outdir\&infile3";
filename infile4 "&outdir\&infile4";
filename infile5 "&outdir\&infile5";
filename infile6 "&outdir\&infile6";
filename infile7 "&outdir\&infile7";
filename infile8 "&outdir\&infile8";
filename infile9 "&outdir\&infile9";
filename infile10 "&outdir\&infile10";
filename infile11 "&outdir\&infile11";
filename infile12 "&outdir\&infile12";
filename infile13 "&outdir\&infile13";
filename infile14 "&outdir\&infile14";
filename outdbf "&outdir\&outfile";
libname outdata "&outdir";


** Process the COMPARTMENT-hex intersection theme **
** Input variables:
**    COMPART  -- compartment number
**    SECTION  -- section of compartment
**    GROUP    -- PNWR management group
**    ACRES, AREA, PERIMETER -- usual measurement statistics
**    HEX_ID   -- ID value assigned to hexagon
**    X, Y     -- real coordinates of hexagon center
**    ROW, COL -- grid coordinates of hexagon  **;
proc dbf db4=infile2 out=a2;  run;

proc sort data=a2;  by hex_id descending area;  run;

/*****
proc summary data=a2 nway;
  class hex_id;
  var area;
  output out=b n=n;
  run;
proc univariate data=b;  var n;  id hex_id;  run;
*****/
```

```
** This step determines compartment membership within hexagons **;
data a2 (keep=hex_id c1-c5 dc1-dc5);
  set a2;
  by hex_id;
  retain c1-c5 dc1-dc5 . count 0;
  ** C1-C5 is compartment membership, DC1-DC5 is compartment area
     C1 and DC1 are ID and area of largest compartment member **;
  array comps[2,5] c1-c5 dc1-dc5;
  if first.hex_id then
    do;
      count = 1;
      do i=1 to 2;
        do j=2 to 5;
          comps[i,j] = .;
        end;
      end;
    end;
  else count = count+1;
  comps[1,count] = compart;
  comps[2,count] = area;
  if last.hex_id then output;
  run;

** Process the STAND-hex intersection theme **
** Input variables:
**    COMP     -- compartment number
**    SECTION  -- section of compartment
**    TRTYEAR  -- year of most recent compartment treatment
**    PREVYEAR -- year of 2nd most recent treatment
**    STAND    -- stand identifier (character)
**    STANDNUM -- number of stand within compartment and stand type
**    S_TYPE1  -- stand type (9 classes, character)
**    S_TYPE2  -- stand type (5 classes, character)
**    COVCLASS -- cover class for pine stands
**    P1YEAR   -- regeneration year of P1 stand
**    AGE, BA  -- age and basal area of pine stand, if known
**    ACRES, AREA, PERIMETER -- usual measurement statistics
**    HEX_ID   -- ID value assigned to hexagon
**    X, Y     -- real coordinates of hexagon center
**    ROW, COL -- grid coordinates of hexagon  **;
proc dbf db4=infile1 out=a1;  run;

proc sort data=a1;  by hex_id descending area;  run;
```

Appendix C.5. *Hexagon Attributes.SAS*. Continued.

```
** This step outputs a single record per hexagon.  The compartment composition
**   (C1-C5), compartment areas (DC1-DC5), and total stand type areas (D_HWB,
**   D_HWU, D_O, D_W, D_P1, D_P2, D_P3A, D_P3B, and D_P3C) within the hexagon
**   are output.  For pine stands, the age and BA of the largest stand in the
**   hexagon are also output (AGE_P1, AGE_P2, AGE_P3A, AGE_P3B, AGE_P3C, BA_P1,
**   BA_P2, BA_P3A, BA_P3B, BA_P3C), regardless of whether the hexagon
**   straddles a compartment boundary or is completely contained in a
**   compartment.  Also, the designation of most recent treatment year
**   (TRTYEAR) and 2nd most recent treatment year (PREVYEAR) associated with
**   the compartment comprising the majority of the hexagon (C1) are assigned
**   to the hexagon.                                                 ****;
data a1 (keep=hex_id row col x y c1-c5 dc1-dc5 keepyr1 keepyr2
              d_hwb d_hwu d_o d_w d_p1 d_p2 d_p3a d_p3b d_p3c
              age_p1 age_p2 age_p3a age_p3b age_p3c
              ba_p1 ba_p2 ba_p3a ba_p3b ba_p3c
         rename=(keepyr1=trtyear keepyr2=prevyear));
  merge a2 a1;
  by hex_id;
  retain d_hwb d_hwu d_o d_w d_p1 d_p2 d_p3a d_p3b d_p3c keepyr1 keepyr2
         age_p1 age_p2 age_p3a age_p3b age_p3c
         ba_p1 ba_p2 ba_p3a ba_p3b ba_p3c;
  array habs[9] d_hwb d_hwu d_o d_w d_p1 d_p2 d_p3a d_p3b d_p3c;
  array habc[9] $ _temporary_ ('Hd-B' 'Hd-U' 'Open' 'Water' 'P1' 'P2'
                               'P3A' 'P3B' 'P3C');

  **  Age and BA of stands in hexagons split over compartment boundaries
  **    are assigned on the basis of the age and BA of the largest stand
  **    within the cell. **;
  array ages[5] age_p1 age_p2 age_p3a age_p3b age_p3c;
  array bas[5] ba_p1 ba_p2 ba_p3a ba_p3b ba_p3c;
  ** Replace missing AGE value in pine stands.  Values based on approx.
  **   size-age relationships in 1982 Habitat Management Plan (p. 85).;
  if age=-3 then
    do;
      if s_type1 = 'P2'  then age = 28;
      if s_type1 = 'P3A' then age = 64;
      if s_type1 = 'P3B' then age = 64;
      if s_type1 = 'P3C' then age = 64;
    end;
  ** Replace missing BA value in pine stands.  Values based roughly on
  **   BA values in measured stands.;
  if ba=-3 then
    do;
```

```
    if s_type1 = 'P2'  then ba = 70;
    if s_type1 = 'P3A' then ba = 85;
    if s_type1 = 'P3B' then ba = 70;
    if s_type1 = 'P3C' then ba = 55;
  end;
** For pine stands, set AGE = YEAR OF ESTABLISHMENT   **;
if s_type1 = 'P1' then age = p1year;
else if substr(s_type1,1,1)='P' then age = trtyear - age;
** Initialize area, age and BA vectors **;
if first.hex_id then
  do;
    do i=1 to 9;
      habs[i] = 0;
    end;
    do i=1 to 5;
      ages[i] = .;
      bas[i] = .;
    end;
    keepyr = .;
  end;
** Augment type-specific area vector **;
do i=1 to 9;
  if s_type1=habc[i] then habs[i]=habs[i]+area;
end;
** Make sure that TRTYEAR is associated with the C1 variable **;
if comp=c1 then
  do;
    keepyr1 = trtyear;
    keepyr2 = prevyear;
  end;
** Assign age and BA of the largest of each of the pine types within a
     hexagon to every stand in the hexagon.  That is, if 2 P3C stands are
     found within a hexagon, then assign age and BA for the larger stand to
     the smaller stand.  For hexagons contained entirely within a compartment,
     P2 and P3 stands already are set to consistent values of age and BA.
     For hexagons that straddle a compartment border, age and BA for a stand
     type are forced to be consistent among similar stands according to the
     age and BA of the largest of the stands.  P1 stands within a compartment
     may be of different ages--this step assures that, within a hexagon, P1
     stands are set to the same age.  **;
do i=1 to 5;
  if s_type1=habc[i+4] then
    do;
```

Appendix C.5. *Hexagon Attributes.SAS*. Continued.

```
        if ages[i]=. then ages[i] = age;
        if bas[i]=. then bas[i] = ba;
      end;
  end;
  if last.hex_id then output;
  run;


** Read in the hex-only theme, save original sort order **;
proc dbf db4=infile3 out=b;  run;

data b;
  set b;
  sortord = _n_;
  run;

/**
** This FREQ can be run to check the distribution of areas on the hexagon
      theme (all interior hexagons should be of the same size, but are not) *;
proc freq data=b;
  tables area;
  run;
**/

proc sort data=b;  by hex_id;  run;

**  This step makes hexagon areas consistent (hexagons created by ARCView
**    vary slightly in area), with corresponding proportional adjustments to
**    areas of stands contained in each hexagon.  Also, pine stands are
**    aggregated from 5 classes into the two most dominant classes in the
**    hexagon.  Total pine area is allocated into these 2 classes in proportion
**    to the size of each class.;
data b (keep=hex_id sortord row col x y hexarea outside
             d_hwb d_hwu d_o d_w c1-c5 dc1-dc5 trtyear prevyear
               d_p1   d_p2   d_p3a   d_p3b   d_p3c
             agg_p1 agg_p2 agg_p3a agg_p3b agg_p3c harea1-harea5 hab1-hab5
             age_p1 age_p2 age_p3a age_p3b age_p3c
              ba_p1  ba_p2  ba_p3a  ba_p3b  ba_p3c);
  merge b a1;
  by hex_id;
  ** If one of the interior hexagons, amount of outside area=0 **;
  if (&&&int) then outside = 0;
  ** If not an interior hexagon, check if hexagon is intermediate in size
    between smallest and largest interior hexagons.  If it is, then let
    outside area be the difference in size from the largest hexagon **;
else if &&&minint<area<&&&maxint then outside = &&&maxint - area;
** If it is not, let outside area be the difference in size from the
    target hexagon size **;
else outside = &&&avgint - area;
** Store the target hexagon size in variable HEXAREA **;
hexarea = &&&avgint;
** Adjust all habitat areas so that the hexagon sum is now
     HEXAREA - OUTSIDE (target hexagon size less outside area) **;
array habs[9] d_hwb d_hwu d_o d_w d_p1 d_p2 d_p3a d_p3b d_p3c;
do i=1 to 9;
  habs[i] = habs[i]*(hexarea-outside)/area;
end;
** SORTAREA array stores the ordered pine class areas
   SORTHABS array stores the pine class indicators for the sorted list
   AGGHABS array stores aggregated pine class areas (2 aggregated classes)
   TEMPHABS is a temporary array  **;
array sortarea[5] harea1-harea5;
array sorthabs[5] hab1-hab5;
array temphabs[5] tarea1-tarea5;
array agghabs[5] agg_p1 agg_p2 agg_p3a agg_p3b agg_p3c;
** Initialize temporary and aggregation arrays **;
do i=1 to 5;
  temphabs[i] = habs[i+4];
  agghabs[i] = 0;
end;
** Determine rank order of pine class areas **;
do i=1 to 5;
  maxi = -1;
  tindex = 0;
  do j=1 to 5;
    if temphabs[j]>=maxi then
      do;
        tindex = j;
        maxi = temphabs[j];
      end;
  end;
  sorthabs[i] = tindex;
  sortarea[i] = maxi;
  temphabs[tindex] = -100;
end;
** Aggregate habitats only if any pine habitat exists in hexagon **;
```

Appendix C.5. *Hexagon Attributes.SAS*. Continued.

```
  if sortarea[1]>0 then
    do;
      p1 = sortarea[1]/(sortarea[1]+sortarea[2]);
      p2 = 1-p1;
      agghabs[hab1] = sortarea[1] + p1*(sortarea[3]+sortarea[4]+sortarea[5]);
      agghabs[hab2] = sortarea[2] + p2*(sortarea[3]+sortarea[4]+sortarea[5]);
    end;
  run;

** Read in the hexagon treatment theme **
** Input variables:
**    HEX_ID   -- ID value assigned to hexagon
**    ROW, COL -- grid coordinates of hexagon
**    OUTSIDE  -- difference between target and actual hexagon size
**    C1       -- compartment number
**    TRTYEAR  -- most recent compartment treatment year
**    B1991 - B2000  -- indicators of annual burns in hexagons
**    LAST     -- year of most recent burn in hexagon
**    SUM      -- total burns in hexagon 1991-2000
**    LASTTHIN -- indicator of thinning in last treatment cycle *;
proc dbf db4=infile4 out=a1;  run;

proc sort data=a1;  by hex_id;  run;

** Read in the hexagon site index theme **
** Input variables:
**    HEX_ID   -- ID value assigned to hexagon
**    COUNT    -- number of site index pixels in hexagon
**    AREA     -- hexagon area
**    MIN, MAX, RANGE, MEAN, STD, SUM
**             -- descriptive statistics of SI values in hexagon
**    VARIETY  -- number of unique SI values in hexagon
**    MAJORITY -- mode of SI values in hexagon (ties?)
**    MINORITY -- least-encountered SI value in hexagon (highest value of ties?)
**    MEDIAN   -- median of SI values in hexagon **;
proc dbf db4=infile5 out=a2;  run;

proc sort data=a2;  by hex_id;  run;

** Read in the hexagon slope theme **
** Input variables:
**    HEX_ID   -- ID value assigned to hexagon
**    COUNT    -- number of slope pixels in hexagon
**    AREA     -- hexagon area
**    MIN, MAX, RANGE, MEAN, STD, SUM
**             -- descriptive statistics of slope values in hexagon
**    VARIETY  -- number of unique slope values in hexagon
**    MAJ1     -- mode of slope values in hexagon (ties?)
**    MIN1     -- least-encountered slope value (highest value of ties?)
**    MED1     -- median of slope values in hexagon **;
proc dbf db4=infile6 out=a3;  run;

proc sort data=a3;  by hex_id;  run;

** Read in the hexagon slope position theme **
** Input variables:
**    HEX_ID   -- ID value assigned to hexagon
**    COUNT    -- number of slope position pixels in hexagon
**    AREA     -- hexagon area
**    MIN, MAX, RANGE, MEAN, STD, SUM
**             -- descriptive statistics of position values in hexagon
**    VARIETY  -- number of unique position values in hexagon
**    MAJORITY -- mode of position values in hexagon (ties?)
**    MINORITY -- least-encountered position value (highest value of ties?)
**    MEDIAN   -- median of position values in hexagon **;
proc dbf db4=infile7 out=a4;  run;

proc sort data=a4;  by hex_id;  run;

** Read in the hexagon-stream intersection theme **
** Important input variables:
**    HEX_ID   -- ID value assigned to hexagon
**    HYDRO_ID -- ID value of stream
**    LENGTH   -- length (m) of stream segment **;
proc dbf db4=infile8 out=a5;  run;

** Add length values over all streams within hexagon **;
proc summary data=a5 nway;
  class hex_id;
  var length;
  output out=a5 sum=s_dens;
  run;

** Read in the distances theme (distances from hexagon centroids) **
** Important input variables:
**    HEX_ID   -- ID value assigned to hexagon
```

Appendix C.5. *Hexagon Attributes.SAS*. Continued.

```
**    D_STREAM -- Distance to nearest stream feature
**    D_PNWR   -- Distance to PNWR border (=0 for centroids inside PNWR)
**    D_ONF    -- Distance to nearest Oconee NF border
**    D_P_IN   -- Distance to nearest private inholding
**    D_P_OUT  -- Distance to nearest surrounding private land  **;
proc dbf db4=infile9 out=a6 (drop=area);  run;

proc sort data=a6;  by hex_id;  run;

** Read in the land ownership layer intersected with 400-m circular buffers **
** Important input variables:
**    HEX_ID   -- ID value assigned to hexagon
**    OWNER    -- ownership status of polygon
**    AREA     -- area (m^2) of polygon  **;
proc dbf db4=infile10
         out=a7 (keep=hex_id ownership area rename=(ownership=owner));  run;

proc sort data=a7;  by hex_id owner;  run;

** Calculate ownership proportions in 400-m buffers **;
data a7 (drop=i owner area sumarea);
  set a7;
  by hex_id;
  retain b4_pnwr b4_onf b4_pin b4_pout sumarea;
  array own{4} b4_pnwr b4_onf b4_pin b4_pout;
  array ownclass{4} $ _temporary_ ('PNWR' 'ONF' 'Priv-In' 'Priv-Out');
  if first.hex_id then
    do i=1 to 4;
        own[i] = 0;
    end;
  do i=1 to 4;
    if ownclass[i]=owner then own[i] = area;
  end;
  if last.hex_id then
    do;
      sumarea = b4_pnwr+b4_onf+b4_pin+b4_pout;
      do i=1 to 4;
        own[i] = own[i]/sumarea;
      end;
      output;
    end;
  run;
```

```
** Read in the land ownership layer intersected with 800-m circular buffers **
** Important input variables:
**    HEX_ID   -- ID value assigned to hexagon
**    OWNER    -- ownership status of polygon
**    AREA     -- area (m^2) of polygon  **;
proc dbf db4=infile11
         out=a8 (keep=hex_id ownership area rename=(ownership=owner));  run;

proc sort data=a8;  by hex_id owner;  run;

** Calculate ownership proportions in 800-m buffers **;
data a8 (drop=i owner area sumarea);
  set a8;
  by hex_id;
  retain b8_pnwr b8_onf b8_pin b8_pout sumarea;
  array own{4} b8_pnwr b8_onf b8_pin b8_pout;
  array ownclass{4} $ _temporary_ ('PNWR' 'ONF' 'Priv-In' 'Priv-Out');
  if first.hex_id then
    do i=1 to 4;
        own[i] = 0;
    end;
  do i=1 to 4;
    if ownclass[i]=owner then own[i] = area;
  end;
  if last.hex_id then
    do;
      sumarea = b8_pnwr+b8_onf+b8_pin+b8_pout;
      do i=1 to 4;
        own[i] = own[i]/sumarea;
      end;
      output;
    end;
  run;

** Read in the land ownership layer intersected with 1600-m circular buffers **
** Important input variables:
**    HEX_ID   -- ID value assigned to hexagon
**    OWNER    -- ownership status of polygon
**    AREA     -- area (m^2) of polygon  **;
proc dbf db4=infile12
         out=a9 (keep=hex_id ownership area rename=(ownership=owner));  run;

proc sort data=a9;  by hex_id owner;  run;
```

Appendix C.5. *Hexagon Attributes.SAS*. Continued.

```
** Calculate ownership proportions in 1600-m buffers **;
data a9 (drop=i owner area sumarea);
  set a9;
  by hex_id;
  retain b16_pnwr b16_onf b16_pin b16_pout sumarea;
  array own{4} b16_pnwr b16_onf b16_pin b16_pout;
  array ownclass{4} $ _temporary_ ('PNWR' 'ONF' 'Priv-In' 'Priv-Out');
  if first.hex_id then
    do i=1 to 4;
        own[i] = 0;
    end;
  do i=1 to 4;
    if ownclass[i]=owner then own[i] = area;
  end;
  if last.hex_id then
    do;
      sumarea = b16_pnwr+b16_onf+b16_pin+b16_pout;
      do i=1 to 4;
        own[i] = own[i]/sumarea;
      end;
      output;
    end;
  run;

** Read in the land ownership layer intersected with 3200-m circular buffers **
** Important input variables:
**    HEX_ID   -- ID value assigned to hexagon
**    OWNER    -- ownership status of polygon
**    AREA     -- area (m^2) of polygon  **;
proc dbf db4=infile13
         out=a10 (keep=hex_id ownership area rename=(ownership=owner));  run;

proc sort data=a10;  by hex_id owner;  run;

** Calculate ownership proportions in 3200-m buffers **;
data a10 (drop=i owner area sumarea);
  set a10;
  by hex_id;
  retain b32_pnwr b32_onf b32_pin b32_pout sumarea;
  array own{4} b32_pnwr b32_onf b32_pin b32_pout;
  array ownclass{4} $ _temporary_ ('PNWR' 'ONF' 'Priv-In' 'Priv-Out');
  if first.hex_id then
    do i=1 to 4;
        own[i] = 0;
    end;
  do i=1 to 4;
    if ownclass[i]=owner then own[i] = area;
  end;
  if last.hex_id then
    do;
      sumarea = b32_pnwr+b32_onf+b32_pin+b32_pout;
      do i=1 to 4;
        own[i] = own[i]/sumarea;
      end;
      output;
    end;
  run;

** Read in the year 2000 RCW population status **
** Important input variables:
**    HEX_ID   -- ID value assigned to hexagon
**    RCW2000  -- Active/inactive status
**    CLUSTYPE -- Cluster type is active (A) or recruitment (R) **;
proc dbf db4=infile14 out=a11 (rename=(clus_type=clustype));  run;

proc sort data=a11;  by hex_id;  run;



**  This step merges the files of hexagon stand attributes, hexagon treatment
**    histories, and statistics on site index, slope, slope position,
**    stream length, distance, and ownership portions.  The pine stand
**    information is saved in two sets of variables representing the first
**    and second-dominating types.  **;
data b (keep=hex_id sortord row col x y hexarea outside inside
             d_hwb d_hwu d_o d_w c1-c5 trtyear prevyear
             ptype1 ptype2 area1 area2 ba1 ba2 age1 age2 cc1 cc2 sumarea
             burnhist last lastthin tsburn tsthin
             i_n i_min i_max i_rng i_mean i_std i_vals i_mode i_medn
             s_n s_min s_max s_rng s_mean s_std s_vals s_mode s_medn
             p_n p_min p_max p_rng p_mean p_std p_vals p_mode p_medn
             s_dens d_stream d_pnwr d_onf d_p_in d_p_out
             b4_pnwr b4_onf b4_pin b4_pout
             b8_pnwr b8_onf b8_pin b8_pout
             b16_pnwr b16_onf b16_pin b16_pout
             b32_pnwr b32_onf b32_pin b32_pout
```

Appendix C.5. *Hexagon Attributes.SAS*. Continued.

```
           rcw2000 clustype
      rename=(last=lastburn));
retain hex_id sortord row col x y hexarea outside inside c1-c5
           trtyear prevyear d_hwb d_hwu d_o d_w
           ptype1 area1 age1 ba1 cc1 ptype2 area2 age2 ba2 cc2 sumarea
           burnhist last lastthin tsburn tsthin
           i_n i_min i_max i_rng i_mean i_std i_vals i_mode i_medn
           s_n s_min s_max s_rng s_mean s_std s_vals s_mode s_medn
           p_n p_min p_max p_rng p_mean p_std p_vals p_mode p_medn
           s_dens d_stream d_pnwr d_onf d_p_in d_p_out
           b4_pnwr b4_onf b4_pin b4_pout
           b8_pnwr b8_onf b8_pin b8_pout
           b16_pnwr b16_onf b16_pin b16_pout
           b32_pnwr b32_onf b32_pin b32_pout
           rcw2000 clustype;
merge b
      a1 (keep=hex_id b1991-b2000 last lastthin)
      a2 (drop=area sum minority
          rename=(count=i_n min=i_min max=i_max range=i_rng mean=i_mean
                  std=i_std variety=i_vals majority=i_mode median=i_medn) )
      a3 (drop=area sum min1
          rename=(count=s_n min=s_min max=s_max range=s_rng mean=s_mean
                  std=s_std variety=s_vals maj1=s_mode med1=s_medn) )
      a4 (drop=area sum minority
          rename=(count=p_n min=p_min max=p_max range=p_rng mean=p_mean
                  std=p_std variety=p_vals majority=p_mode median=p_medn) )
      a5 (keep=hex_id s_dens)
      a6 a7 a8 a9 a10
      a11 (keep=hex_id rcw2000 clustype);
by hex_id;
inside = hexarea-outside;
if s_dens<0 then s_dens = 0;
s_dens = s_dens/inside;
format d_hwb d_hwu d_o d_w area1 area2 outside hexarea 10.8
       ba1 ba2 age1 age2 cc1 cc2 6.2;
length burnhist $ 10;
burnhist = put(b1991,1.) || put(b1992,1.) || put(b1993,1.) ||
           put(b1994,1.) || put(b1995,1.) || put(b1996,1.) ||
           put(b1997,1.) || put(b1998,1.) || put(b1999,1.) ||
           put(b2000,1.);
array habs[5] agg_p1 agg_p2 agg_p3a agg_p3b agg_p3c;
array ages[5] age_p1 age_p2 age_p3a age_p3b age_p3c;
array bas[5] ba_p1 ba_p2 ba_p3a ba_p3b ba_p3c;
```

```
  array ccs[5] _temporary_ (100 70 100 70 40);
  array labels[5] $ _temporary_ ('P1' 'P2' 'P3A' 'P3B' 'P3C');
  length ptype1 ptype2 $ 3;
  ptype1 = '   ';
  ptype2 = '   ';
  area1 = 0;
  area2 = 0;
  ba1 = -1;
  ba2 = -1;
  age1 = -1;
  age2 = -1;
  cc1 = -1;
  cc2 = -1;
  if harea1>0 then
    do;
      ptype1 = labels[hab1];
      area1 = habs[hab1];
      ba1 = bas[hab1];
      age1 = ages[hab1];
      cc1 = ccs[hab1];
    end;
  if harea2>0 then
    do;
      ptype2 = labels[hab2];
      area2 = habs[hab2];
      ba2 = bas[hab2];
      age2 = ages[hab2];
      cc2 = ccs[hab2];
    end;
  sumarea = sum(d_hwb,d_o,d_w,outside,d_hwu,area1,area2);
  ** If not thinned in last management cycle, set thinning date to PREVYEAR *;
  if lastthin=0 then lastthin = prevyear;
  ** Time since burn, time since thin *;
  tsburn = 2000-last;
  tsthin = 2000-lastthin;
  run;

/****
**  Create a DBF file for use in ARCView **;
proc sort data=b out=a1 (drop=sortord sumarea);  by sortord;  run;

proc dbf db4=outdbf data=a1;  run;
****/
```

Appendix C.5. *Hexagon Attributes.SAS*. Continued.

```
**  Create permanent SAS dataset **;
data outdata.pnwr_hex;
  set b;
  drop sortord sumarea;
  run;

options ls=80;
proc contents data=outdata.pnwr_hex;  run;

**  List hexagons by increasing size **;
proc sort data=b;
  by inside;
  run;

data a1;
  set b;
  retain areasum 0;
  areasum+inside;
  run;

options ls=120;
proc print data=a1;
  var hex_id inside areasum i_n i_medn s_n s_medn p_n p_medn s_dens
            /*burnhist*/;
  format hex_id 8. i_medn s_medn 8.4 s_dens 12.8;
  run;

options ls=80;
/**
proc freq data=b;
  table inside;
  run;
**/

/**
proc print data=b;
  id hex_id;
  var c1 trtyear d_hwb d_o d_w outside d_hwu ptype1 area1 ptype2 area2;
  run;

proc print data=b;
  id hex_id;
  var c1 trtyear ptype1 area1 ba1 age1 cc1 ptype2 area2 ba2 age2 cc2;
  run;

proc univariate data=b;
  var sumarea;
  run;
**/

  quit;
```

Appendix C.6. *Advance cells to current.SAS*. Reads SAS file of hexagon attributes PNWR_HEX and advances the forest state variables for pine stands (basal area and canopy closure) to a single point in time, specified in the YEARNOW macro variable. Program uses basal area and canopy closure models as described in report. Output is SAS file PNWRHAB.

```
**  Advance cells to current.SAS  **
**  This program simulates growth of the forest and understory according
     to the Bailey-Ware basal area growth model.
     The forest is brought forward to a common point in time from compartment-
     specific treatment dates. **;

    ** Parameters for Bailey-Ware model **;
%let thinindx = 0.6;   ** Values <1.0 represent degree of thinning from below;
%let b1 = 2.81706;     **;
%let b2 = -11935.2;    ** Model coefficients **;
%let b3 = 0.043493;    **;
%let age0 = 8;         ** Age at which BA follows Bailey-Ware model **;
%let ba0 = 0.46;    ** Constant BA (2.00 ft^2/ac) value up through age AGE0 **;
%let ba_max = 35;   ** Maximum BA (152 ft^2/ac) allowed.  BA model is
                       sensitive to errors introduced by discretization of
                       stands and treatment years into hexagons, especially
                       for P2 stands.    ***;

    ** Parameter for canopy closure model **;
%let ccrate = 0.79;

%let yearnow = 2000;
%let workpath = d:\projects\refuge arm\refuge model;
%let datapath = d:\pnwrmap\hex10;
%let outpath = d:\projects\refuge arm\refuge model;

libname inforest "&datapath";
libname outfor "&outpath";

data outfor.pnwrhab;
  set inforest.pnwr_hex
        (keep = hex_id row col x y outside inside c1-c2 trtyear prevyear
                lastburn lastthin d_hwb d_hwu d_o d_w area1 age1 ba1 cc1
                area2 age2 ba2 cc2 i_medn s_medn p_medn
                s_dens d_stream d_pnwr d_onf d_p_in d_p_out
                b4_pnwr b4_onf b4_pin b4_pout
                b8_pnwr b8_onf b8_pin b8_pout
                b16_pnwr b16_onf b16_pin b16_pout
                b32_pnwr b32_onf b32_pin b32_pout
              rcw2000 clustype);
** Exclude hexagons < 1/4 acre in size **;
if inside>=1011.714;
** Set parameter values **;
x0 = (&thinindx^=0)*(1-&thinindx);       ** thinning index **;
b1 = &b1;
b2 = &b2;
b3 = &b3;
age0 = &age0;
ba0 = &ba0;
ba_max = &ba_max;
ccrate = &ccrate;
si_medn = i_medn/3.280839895; * Median of site index (m) for cell *;

** Calculation of contemporary basal area and canopy closure for FIRST
     component **;
ba11 = ba1/4.356;             ** BA (m^2/ha) at treatment year **;
** AGE is regeneration year, AGE = -1 when missing **;
if age1>0 then a11 = trtyear - age1;
else a11 = age1;
a1_t = a11 - (trtyear-lastthin);           ** Age at last thinning **;
** Change age, age at last thin, BA to missing if component not present
    (missing component => AGE<0 and BA<0).
    Also, age at last thin is missing if stand is P1 (AGE>-1 & BA<=0)  **;
if a11<0 then a11 = .;
if ba11<0 then ba11 = .;
if a11<0 | (a11>-1 & ba11<=0) then a1_t = .;
** Set thinning index based on whether ever thinned **;
if a11>. & a1_t<0 then x01 = 0;
else if a11>. then x01 = x0;
a12 = a11 + (&yearnow - trtyear);  ** age at YEARNOW **;
** P1 stands **;
if a11>. & ba11=. then
  do;
    ** Next age is greater than age threshold **;
    if age0<a12 then
      do;
        arat = age0/a12;
        ba1 = (ba0**arat) * exp( b1*(1-arat) + b3*si_medn*(1-arat) );
```

Appendix C.6. *Advance cells to current.SAS*. Continued.

```
      end;
    else ba1 = ba0;
    ** Canopy closure is function of next age **;
    cc1 = 100 - 100*ccrate**a12;
  end;
else
  do;
    arat = a11/a12;
    prod = exp( b1*(1-arat) + b2*x01*(1/a12-1/a11)/a1_t/a12
              + b3*si_medn*(1-arat) );
    ba1 = (ba11**(arat))*prod;
    ** Canopy closure is function of existing closure and time interval **;
    cc1 = 100 - (100-cc1)*ccrate**(a12-a11);
  end;
** Set maximum BA for P2 stands **;
if 16<=a12<=40 then ba1 = min(ba1,ba_max);
age1 = a12;

** Calculation of contemporary basal area and canopy closure for SECOND
     component **;
ba21 = ba2/4.356;             ** BA (m^2/ha) at treatment year **;
** AGE is regeneration year, AGE = -1 when missing **;
if age2>0 then a21 = trtyear - age2;
else a21 = age2;
a2_t = a21 - (trtyear-lastthin);           ** Age at last thinning **;
** Change age, age at last thin, BA to missing if component not present
    (missing component => AGE<0 and BA<0).
    Also, age at last thin is missing if stand is P1 (AGE>-1 & BA<=0)  **;
if a21<0 then a21 = .;
if ba21<0 then ba21 = .;
if a21<0 | (a21>-1 & ba21<=0) then a2_t = .;
** Set thinning index based on whether ever thinned **;
if a21>. & a2_t<0 then x02 = 0;
else if a21>. then x02 = x0;
a22 = a21 + (&yearnow - trtyear);  ** age at YEARNOW **;
** P1 stands **;
if a21>. & ba21=. then
  do;
    ** Next age is greater than age threshold **;
    if age0<a22 then
      do;
        arat = age0/a22;
        ba2 = (ba0**arat) * exp( b1*(1-arat) + b3*si_medn*(1-arat) );
```

```
        end;
      else ba2 = ba0;
      ** Canopy closure is function of next age **;
      cc2 = 100 - 100*ccrate**a22;
    end;
  else
    do;
      arat = a21/a22;
      prod = exp( b1*(1-arat) + b2*x02*(1/a22-1/a21)/a2_t/a22
                + b3*si_medn*(1-arat) );
      ba2 = (ba21**(arat))*prod;
      ** Canopy closure is function of existing closure and time interval **;
      cc2 = 100 - (100-cc2)*ccrate**(a22-a21);
    end;
  ** Set maximum BA for P2 stands **;
  if 16<=a22<=40 then ba2 = min(ba2,ba_max);
  age2 = a22;

  yearnow = &yearnow;
  run;

/***  Use these steps to calculate maximum BA in P2 stands in hexagons
        contained entirely in a single compartment                      **
proc summary data=outfor.pnwrhab (where=(16<=age1<=40 & c2=.)) nway;
  var ba1;
  output out=a1 max=max_ba1;
  run;

proc summary data=outfor.pnwrhab (where=(16<=age2<=40 & c2=.)) nway;
  var ba2;
  output out=a2 max=max_ba2;
  run;

data a1;
  merge a1 (keep=max_ba1) a2 (keep=max_ba2);
  run;

proc print data=a1;  run;
****/

proc univariate data=outfor.pnwrhab;
  var ba1 ba2 cc1 cc2;
  run;  quit;
```

Appendix C.7. *Export of hex data to GAUSS.SAS*. SAS program splits the SAS database PNWRHAB into two portions and writes each to Excel files for import into GAUSS. Excel files created are HEX_FOREST.XLS and HEX_ATTRIB.XLS.

```
libname inforest "d:\projects\refuge arm\refuge model";

data a;
  keep   hex_id row col x y inside
         c1 trtyear prevyear lastburn lastthin
         d_hwb d_hwu d_o d_w
         area1 age1 ba1 cc1 area2 age2 ba2 cc2
         i_medn p_medn s_dens d_stream d_pnwr d_onf d_p_in d_p_out
         b4_pnwr b4_onf b4_pin b4_pout
         b8_pnwr b8_onf b8_pin b8_pout
         b16_pnwr b16_onf b16_pin b16_pout
         b32_pnwr b32_onf b32_pin b32_pout
         rcw2000 cluster;
  retain hex_id row col x y inside
         c1 trtyear prevyear lastburn lastthin
         d_hwb d_hwu d_o d_w
         area1 age1 ba1 cc1 area2 age2 ba2 cc2
         i_medn p_medn s_dens d_stream d_pnwr d_onf d_p_in d_p_out
         b4_pnwr b4_onf b4_pin b4_pout
         b8_pnwr b8_onf b8_pin b8_pout
         b16_pnwr b16_onf b16_pin b16_pout
         b32_pnwr b32_onf b32_pin b32_pout
         rcw2000 cluster;
  set inforest.pnwrhab;
  if clustype='' then cluster = 0;
  else if clustype='A' then cluster = 1;
  else if clustype='R' then cluster = 2;
  run;

data a1;
  set a;
  keep hex_id row col x y inside c1 trtyear prevyear lastburn lastthin
         d_hwb d_hwu d_o d_w area1 age1 ba1 cc1 area2 age2 ba2 cc2
         rcw2000 cluster;
  run;

proc export data=a1
            outfile= "d:\projects\refuge arm\refuge model\hex_forest.xls"
            dbms=excel5 replace;
  run;
```

```
data a1;
  set a;
  keep hex_id i_medn p_medn s_dens d_stream d_pnwr d_onf d_p_in d_p_out
         b4_pnwr b4_onf b4_pin b4_pout b8_pnwr b8_onf b8_pin b8_pout
         b16_pnwr b16_onf b16_pin b16_pout b32_pnwr b32_onf b32_pin b32_pout;
  run;

proc export data=a1
            outfile= "d:\projects\refuge arm\refuge model\hex_attrib.xls"
            dbms=excel5 replace;
  run;

quit;
```

Appendix C.8. *Import hex data to GAUSS.G*. GAUSS program reads the two forest attribute Excel files and creates corresponding files HEX_FOREST and HEX_ATTRIB GAUSS matrix files.

```
@ Reads the hexagon attribute data, which was output by SAS in Excel 5
  spreadsheet format.  The data are saved in two files due to large file size:
     HEX_FOREST:  hexagon indices, compartment number, treatment data,
                  forest cover types and cover info, RCW population info
     HEX_ATTRIB:  site index, slope position, stream density, distance data,
                  ownership portions.
  The data and column headers are saved as GAUSS
  matrix files.    @

new;

@ Output file path @
pathname = "d:\\projects\\refuge arm\\refuge model";

@ Input file name for forest data @
fname = "d:\\projects\\refuge arm\\refuge model\\hex_forest.xls";

@ Columns, width, and precision in HEX_FOREST file @
hdr_for  = {hex_id     4  0,
            row        2  0,
            col        2  0,
            x         11  3,
            y         11  3,
            inside     9  3,
            c1         2  0,
            trtyear    4  0,
            prevyear   4  0,
            lastburn   4  0,
            lastthin   4  0,
            d_hwb     11  3,
            d_hwu     11  3,
            d_o       11  3,
            d_w       11  3,
            area1     11  3,
            age1       4  0,
            ba1        7  4,
            cc1        7  4,
            area2     11  3,
            age2       4  0,
            ba2        7  4,
            cc2        7  4,
```

```
            rcw2000    1  0,
            clustype   1  0};

@ Global variables for IMPORT procedure @
_dxwidth = hdr_for[.,2];
_dxprcn = hdr_for[.,3];
_dxwkshdr = 1;

{hex_for,hex1} = import(fname,0,0);
save path=^pathname hex_for, hdr_for;

@ Test print of part of input matrix @
ptfmt = "*.*lf ";
ptfmt = ptfmt*ones(rows(_dxwidth),1) ~ _dxwidth ~ _dxprcn;
print (rows(hex_for)~cols(hex_for));
call printfm(hex_for[1:10,.],1,ptfmt);
call printfmt(hdr_for[.,1],0);


@ Input file name for hexagon attribute data @
fname = "d:\\projects\\refuge arm\\refuge model\\hex_attrib.xls";

@ Columns, width, and precision in HEX_ATTRIB file @
hdr_attr = {hex_id     4  0,
            i_medn     7  4,
            p_medn     1  0,
            s_dens     8  6,
            d_stream   8  2,
            d_pnwr     8  2,
            d_onf      8  2,
            d_p_in     8  2,
            d_p_out    8  2,
            b4_pnwr    8  6,
            b4_onf     8  6,
            b4_pin     8  6,
            b4_pout    8  6,
            b8_pnwr    8  6,
            b8_onf     8  6,
            b8_pin     8  6,
            b8_pout    8  6,
            b16_pnwr   8  6,
```

Appendix C.8. *Import hex data to GAUSS.G*. Continued.

```
            b16_onf    8  6,
            b16_pin    8  6,
            b16_pout   8  6,
            b32_pnwr   8  6,
            b32_onf    8  6,
            b32_pin    8  6,
            b32_pout   8  6};

@ Global variables for IMPORT procedure @
_dxwidth = hdr_attr[.,2];
_dxprcn = hdr_attr[.,3];
_dxwkshdr = 1;

{hex_attr,hex1} = import(fname,0,0);
save path=^pathname hex_attr, hdr_attr;

@ Test print of part of input matrix @
ptfmt = "*.*lf ";
ptfmt = ptfmt*ones(rows(_dxwidth),1) ~ _dxwidth ~ _dxprcn;
print (rows(hex_attr)~cols(hex_attr));
call printfm(hex_attr[1:10,.],1,ptfmt);
call printfmt(hdr_attr[.,1],0);
```

Appendix C.9. ASDP input files for forest model F0. State dynamics, stage return, and scenario files corresponding to model of intermediate rate of hardwood encroachment.

State Dynamics File

```
#include "sd.h"
/*********************************************************************
 PNWR forest succession model
 version HWD1 -- includes hardwood component (UH) w/encroachment rates,
                 logit expression of state space
   CUR_STATE has 4 elements:  P1, P2, P3, P3RCW
     --> 5th state is UH which is obtained by subtraction <--
     A temporary 6th state (RG) is used to compute additions to the P1 class.
       It is emptied during the transitions.  Because of this additional
       component, the relationship between the temporary vectors and the
       DP vectors (CUR_STATE and NXT_STATE) is
                        CUR_STATE[i] = THIS_F[i+1].
   DEC has 3 elements:  P2 harvest, P3 harvest, P3RCW harvest
   OUTCOME has 2 elements:  overall encroachment and catastrophic loss rates
*********************************************************************/
{
  INT
            /** Indices of temporary state vectors **/
          RG = 0,
          P1 = 1,
          P2 = 2,
          P3 = 3,
          P4 = 4,
          UH = 5,
            /** Indices of decision vector **/
          H_P2 = 0,
          H_P3 = 1,
          H_P4 = 2,
            /** Number of steps corresponds to number of state variable
                increments in scenario file **/
          steps = 11,
            /** Other variables **/
          i,j,k;

  DECIMAL
            /** Smallest non-zero proportion value **/
          base_p = 0.001,
            /** Exponent to linearize relationship between logit and
                proportion **/
          power = 2.5,
            /** Mean rates of hardwood encroachment **/
          e_P1 = 0.0006,
          e_P2 = 0.015,
          e_P3 = 0.015,
          e_P4 = 0.03,
            /** UH to P1 conversion fraction **/
          q = 0.25,
            /** P3_RCW admission rate **/
          p = 0.5,
            /** (inverse) age class transition rates **/
          t21 = 16.,
          t32 = 24.,
          t43 = 40.,
          t14 = 40.,
            /** Realized rates of hardwood encroachment **/
          e1, e2, e3, e4,
            /** Harvest-adjusted rates of hardwood encroachment **/
          f1, f2, f3, f4,
            /** Rates of harvest **/
          d2, d3, d4,
            /** Realized rate of catastrophic loss **/
          c,
            /** Catastrophic loss matrix **/
          C[6][6],
            /** Age class transition matrix **/
          G[6][6],
            /** Hardwood encroachment transition matrix **/
          H[6][6],
            /** Decision matrix **/
          D[6][6],
            /** Other temporary quantities **/
          P[6][6],
          T1[6][6],
          T2[6][6],
          this_f[6],
          next_f[6],
          half, scale, lrat, diff, denom, blogit,
          l1, l2, l3, l4, g1, g2, g3, g4;
```

Appendix C.9. ASDP input files for forest model F0. Continued.

```
/** HALF is the half-way increment **/
half = (steps+1.)/2.;
/** Logit value for smallest non-zero proportion value  (constant in
      denominator is number of components-1)                **/
blogit = log(base_p/(1.-4*base_p));
/** SCALE is the scale parameter for the polynomial curve (paired negation
      operators allow exponentiation to fractional power) **/
scale = -pow(-(1.-half),power)/blogit;

/** Read current state variables and convert them to logits
     (Paired negation operators allow exponentiation to fractional power) **/
diff = cur_state[P1-1]-half;
if (diff < 0) l1 = -pow(-diff,power)/scale;
         else l1 = pow(diff,power)/scale;
diff = cur_state[P2-1]-half;
if (diff < 0) l2 = -pow(-diff,power)/scale;
         else l2 = pow(diff,power)/scale;
diff = cur_state[P3-1]-half;
if (diff < 0) l3 = -pow(-diff,power)/scale;
         else l3 = pow(diff,power)/scale;
diff = cur_state[P4-1]-half;
if (diff < 0) l4 = -pow(-diff,power)/scale;
         else l4 = pow(diff,power)/scale;
denom = 1.+exp(l1)+exp(l2)+exp(l3)+exp(l4);

/** Calculate current forest composition states **/
this_f[RG] = 0;
this_f[P1] = exp(l1)/denom;
this_f[P2] = exp(l2)/denom;
this_f[P3] = exp(l3)/denom;
this_f[P4] = exp(l4)/denom;
this_f[UH] = 1./denom;

/** Set hardwood encroachment rates in proportion to rate for P1 **/
e1 = outcome[0];
e2 = e1 * e_P2/e_P1;
e3 = e1 * e_P3/e_P1;
e4 = e1 * e_P4/e_P1;

/** Set catastrophic loss rate **/
c = outcome[1];

/** Matrix C (catastrophic loss) **/
```

```
  C[0][0] = 1.; C[0][1] = 0.; C[0][2] = 0.;   C[0][3] = 0.;   C[0][4] = 0.;   C[0][5]
= 0.;
  C[1][0] = 0.; C[1][1] = 1.; C[1][2] = c;    C[1][3] = c;    C[1][4] = c;    C[1][5]
= c*q;
  C[2][0] = 0.; C[2][1] = 0.; C[2][2] = 1.-c; C[2][3] = 0.;   C[2][4] = 0.;   C[2][5]
= 0.;
  C[3][0] = 0.; C[3][1] = 0.; C[3][2] = 0.;   C[3][3] = 1.-c; C[3][4] = 0.;   C[3][5]
= 0.;
  C[4][0] = 0.; C[4][1] = 0.; C[4][2] = 0.;   C[4][3] = 0.;   C[4][4] = 1.-c; C[4][5]
= 0.;
  C[5][0] = 0.; C[5][1] = 0.; C[5][2] = 0.;   C[5][3] = 0.;   C[5][4] = 0.;   C[5][5]
= 1.-c*q;

  /** Matrix G (age class transitions) **/
  g1 = 1./t21;
  g2 = 1./t32;
  g3 = 1./t43;
  g4 = 1./t14;

  G[0][0] = 0.; G[0][1] = 0.;     G[0][2] = 0.;     G[0][3] = 0.;         G[0][4] =
0.;     G[0][5] = 0.;
  G[1][0] = 1.; G[1][1] = 1.-g1;  G[1][2] = 0.;     G[1][3] = (1.-p)*g3;  G[1][4] =
g4;     G[1][5] = 0.;
  G[2][0] = 0.; G[2][1] = g1;     G[2][2] = 1.-g2;  G[2][3] = 0.;         G[2][4] =
0.;     G[2][5] = 0.;
  G[3][0] = 0.; G[3][1] = 0.;     G[3][2] = g2;     G[3][3] = 1.-g3;      G[3][4] =
0.;     G[3][5] = 0.;
  G[4][0] = 0.; G[4][1] = 0.;     G[4][2] = 0.;     G[4][3] = p*g3;       G[4][4] =
1.-g4;  G[4][5] = 0.;
  G[5][0] = 0.; G[5][1] = 0.;     G[5][2] = 0.;     G[5][3] = 0.;         G[5][4] =
0.;     G[5][5] = 1.;

  /** Matrix H (hardwood encroachment transitions) **/
  d2 = dec[H_P2];
  d3 = dec[H_P3];
  d4 = dec[H_P4];

  f1 = e1;
  f2 = max( 0, e2-d2);
  f3 = max( 0, e3-d3);
  f4 = max( 0, e4-d4);
```

Appendix C.9. ASDP input files for forest model F0. Continued.

```
  H[0][0] = 1.; H[0][1] = 0.;    H[0][2] = 0.;    H[0][3] = 0.;    H[0][4] = 0.;
H[0][5] = 0.;
  H[1][0] = 0.; H[1][1] = 1.-f1; H[1][2] = 0.;    H[1][3] = 0.;    H[1][4] = 0.;
H[1][5] = 0.;
  H[2][0] = 0.; H[2][1] = 0.;    H[2][2] = 1.-f2; H[2][3] = 0.;    H[2][4] = 0.;
H[2][5] = 0.;
  H[3][0] = 0.; H[3][1] = 0.;    H[3][2] = 0.;    H[3][3] = 1.-f3; H[3][4] = 0.;
H[3][5] = 0.;
  H[4][0] = 0.; H[4][1] = 0.;    H[4][2] = 0.;    H[4][3] = 0.;    H[4][4] = 1.-f4;
H[4][5] = 0.;
  H[5][0] = 0.; H[5][1] = f1;    H[5][2] = f2;    H[5][3] = f3;    H[5][4] = f4;
H[5][5] = 1.;

  /** Matrix D (harvest decisions) **/
  D[0][0] = 1.; D[0][1] = 0.; D[0][2] = d2;    D[0][3] = d3;    D[0][4] = d4;
D[0][5] = 0.;
  D[1][0] = 0.; D[1][1] = 1.; D[1][2] = 0.;    D[1][3] = 0.;    D[1][4] = 0.;
D[1][5] = 0.;
  D[2][0] = 0.; D[2][1] = 0.; D[2][2] = 1.-d2; D[2][3] = 0.;    D[2][4] = 0.;
D[2][5] = 0.;
  D[3][0] = 0.; D[3][1] = 0.; D[3][2] = 0.;    D[3][3] = 1.-d3; D[3][4] = 0.;
D[3][5] = 0.;
  D[4][0] = 0.; D[4][1] = 0.; D[4][2] = 0.;    D[4][3] = 0.;    D[4][4] = 1.-d4;
D[4][5] = 0.;
  D[5][0] = 0.; D[5][1] = 0.; D[5][2] = 0.;    D[5][3] = 0.;    D[5][4] = 0.;
D[5][5] = 1.;

  /** initialize NEXT_F and temporary matrices **/
  for (i=0; i<6; i=i+1)
  {
    next_f[i] = 0.;
    for (j=0; j<6; j=j+1)
    {
      T1[i][j] = 0.;
      T2[i][j] = 0.;
      P[i][j] = 0.;
    }
  }

  /** Compute (C*G) **/
  for (i=0; i<6; i=i+1)
    for (j=0; j<6; j=j+1)
      for (k=0; k<6; k=k+1)
        T1[i][j] = T1[i][j] + C[i][k]*G[k][j];

  /** Compute ((C*G) * H) **/
  for (i=0; i<6; i=i+1)
    for (j=0; j<6; j=j+1)
      for (k=0; k<6; k=k+1)
        T2[i][j] = T2[i][j] + T1[i][k]*H[k][j];

  /** Compute ( ((C*G) * H) ) * D) **/
  for (i=0; i<6; i=i+1)
    for (j=0; j<6; j=j+1)
      for (k=0; k<6; k=k+1)
        P[i][j] = P[i][j] + T2[i][k]*D[k][j];

  /** Advance to next forest composition state **/
  for (i=0; i<6; i=i+1)
    for (k=0; k<6; k=k+1)
      next_f[i] = next_f[i] + P[i][k]*this_f[k];


/*
printf("\nthis %f %f %f %f %f
%f",this_f[0],this_f[1],this_f[2],this_f[3],this_f[4],this_f[5]);
printf("\nnext %f %f %f %f %f
%f",next_f[0],next_f[1],next_f[2],next_f[3],next_f[4],next_f[5]);
printf("\ndecisions %f %f %f",d2,d3,d4);

printf("\n%f %f %f %f %f %f",G[0][0],G[0][1],G[0][2],G[0][3],G[0][4],G[0][5]);
printf("\n%f %f %f %f %f %f",G[1][0],G[1][1],G[1][2],G[1][3],G[1][4],G[1][5]);
printf("\n%f %f %f %f %f %f",G[2][0],G[2][1],G[2][2],G[2][3],G[2][4],G[2][5]);
printf("\n%f %f %f %f %f %f",G[3][0],G[3][1],G[3][2],G[3][3],G[3][4],G[3][5]);
printf("\n%f %f %f %f %f %f",G[4][0],G[4][1],G[4][2],G[4][3],G[4][4],G[4][5]);
printf("\n%f %f %f %f %f %f",G[5][0],G[5][1],G[5][2],G[5][3],G[5][4],G[5][5]);
*/

  /** Convert logit values to state values **/
/*printf("\ngot here P1 %f %f",next_f[P1],next_f[UH]);*/
  lrat = log(next_f[P1]/next_f[UH]);
  if (lrat<0) nxt_state[P1-1] = -pow((-lrat*scale),(1./power)) + half;
         else nxt_state[P1-1] = pow((lrat*scale),(1./power)) + half;
/*printf("\ngot here P2 %f %f",next_f[P2],next_f[UH]);*/
  lrat = log(next_f[P2]/next_f[UH]);
  if (lrat<0) nxt_state[P2-1] = -pow((-lrat*scale),(1./power)) + half;
```

Appendix C.9. ASDP input files for forest model F0. Continued.

```
         else nxt_state[P2-1] = pow((lrat*scale),(1./power)) + half;
/*printf("\ngot here P3 %f %f",next_f[P3],next_f[UH]);*/
  lrat = log(next_f[P3]/next_f[UH]);
  if (lrat<0) nxt_state[P3-1] = -pow((-lrat*scale),(1./power)) + half;
         else nxt_state[P3-1] = pow((lrat*scale),(1./power)) + half;
/*printf("\ngot here P4 %f %f",next_f[P4],next_f[UH]);*/
  lrat = log(next_f[P4]/next_f[UH]);
  if (lrat<0) nxt_state[P4-1] = -pow((-lrat*scale),(1./power)) + half;
         else nxt_state[P4-1] = pow((lrat*scale),(1./power)) + half;

  return;
}
```

Stage Return File

```
#include "obj.h"
/**********************************************************************
 PNWR forest succession model
 Stage return calculation
 version HWD1 -- includes hardwood component (UH) w/encroachment rates,
                 logit expression of state space
   STATE has 4 elements:  P1, P2, P3, P3RCW
     --> 5th state is UH which is obtained by subtraction <--
**********************************************************************/
{
  INT
            /** Indices of state vector **/
          P1 = 0,
          P2 = 1,
          P3 = 2,
          P4 = 3,
          UH = 4,
            /** Number of steps corresponds to number of state variable
                increments in scenario file **/
          steps = 11;

  DECIMAL
            /** Smallest non-zero proportion value **/
          base_p = 0.001,
          /** Exponent to linearize relationship between logit and
              proportion **/
        power = 2.5,
        P3R_threshold = 0.030961749,
          /** Other temporary quantities **/
        f[5],
        half, scale, diff, l1, l2, l3, l4, denom, blogit;

/** HALF is the half-way increment **/
half = (steps+1.)/2.;
/** Logit value for smallest non-zero proportion value  (constant in
      denominator is number of components-1)                **/
blogit = log(base_p/(1.-4*base_p));
/** SCALE is the scale parameter for the polynomial curve (paired negation
      operators allow exponentiation to fractional power) **/
scale = -pow(-(1.-half),power)/blogit;

/** Read current state variables and convert them to logits
     (Paired negation operators allow exponentiation to fractional power) **/
diff = cur_state[P1]-half;
if (diff < 0) l1 = -pow(-diff,power)/scale;
         else l1 = pow(diff,power)/scale;
diff = cur_state[P2]-half;
if (diff < 0) l2 = -pow(-diff,power)/scale;
         else l2 = pow(diff,power)/scale;
diff = cur_state[P3]-half;
if (diff < 0) l3 = -pow(-diff,power)/scale;
         else l3 = pow(diff,power)/scale;
diff = cur_state[P4]-half;
if (diff < 0) l4 = -pow(-diff,power)/scale;
         else l4 = pow(diff,power)/scale;
denom = 1.+exp(l1)+exp(l2)+exp(l3)+exp(l4);

/** Calculate current forest composition states **/
f[P1] = exp(l1)/denom;
f[P2] = exp(l2)/denom;
f[P3] = exp(l3)/denom;
f[P4] = exp(l4)/denom;
f[UH] = 1./denom;

/** Amount of P3_RCW (P4) habitat must exceed threshold value.  If it does,
      return is 1000+(habitat-threshold).  If it does not, return is 0. **/
if (f[P4]<P3R_threshold) result = 0.;
```

Appendix C.9. ASDP input files for forest model F0. Continued.

```
  else result = 1000. + (f[P4]-threshold);

 return  (result);
}
```

Scenario File

```
!--------------------------------------------------------------------
! PNWR forest succession model -- version HWD1
!--------------------------------------------------------------------
MAX;
TITLE   ">>> PNWR forest succession model (HWD1) <<<"   ;
!REPORT_ALL      ;
!TRACE ;
ITERATIONS   20|500;
!ITERATIONS 1;
NO_CHANGE 3;

! Simulation settings
! Initial forest state is (P1=0.1227 P2=0.0741 P3=0.6173 P3R=0.001 UH=0.1849)
INIT_STATE  4.3838546  3.7726398  8.4877613  1.5290676;
SIM_TRIALS 1000;
SIM_REPORT 1 1 1 1 1 1 1;
NO_INTERP;

STATE   "P1 index";
STATE   "P2 index";
STATE   "P3 index";
STATE   "P3R index";

DECISION "P2 harvest";
DECISION "P3 harvest";
DECISION "P3R harvest";

DISTRIB "Hdwd P1";
RV "Hdwd P1";
!EVENT 1.0 0.0006;

EVENT      0.1    0.00025589412;
EVENT      0.2    0.00039234709;
EVENT      0.4    0.00057114825;
EVENT      0.2    0.00079435978;
EVENT      0.1    0.0010860989;

DISTRIB "Cat rate";
RV "Cat rate";
!EVENT 1.0 0.003394;

!EVENT   0.500   6.7989136e-014;
EVENT   0.500   0.000000000000067989136;
EVENT   0.400   0.00011697715;
EVENT   0.050   0.0058095462;
EVENT   0.045   0.045113811;
EVENT   0.005   0.20532215;

STAGE 1 1 1 1;
COMBINE 1|11 1|11 1|11 1|11,
        0|0.14|0.02 0|0.14|0.02 0|0.14|0.02,
        "Hdwd P1" "Cat rate";
```

Appendix C.10. ASDP input files for forest model F1. State dynamics, stage return, and scenario files corresponding to model of rapid rate of hardwood encroachment.

State Dynamics File

```
#include "sd.h"
/*********************************************************************
 PNWR forest succession model (sensitivity analysis version ALT1)
 version HWD1 -- includes hardwood component (UH) w/encroachment rates,
                 logit expression of state space
   CUR_STATE has 4 elements:  P1, P2, P3, P3RCW
     --> 5th state is UH which is obtained by subtraction <--
     A temporary 6th state (RG) is used to compute additions to the P1 class.
       It is emptied during the transitions.  Because of this additional
       component, the relationship between the temporary vectors and the
       DP vectors (CUR_STATE and NXT_STATE) is
                        CUR_STATE[i] = THIS_F[i+1].
   DEC has 3 elements:  P2 harvest, P3 harvest, P3RCW harvest
   OUTCOME has 2 elements:  overall encroachment and catastrophic loss rates
*********************************************************************/
{
  INT
            /** Indices of temporary state vectors **/
          RG = 0,
          P1 = 1,
          P2 = 2,
          P3 = 3,
          P4 = 4,
          UH = 5,
            /** Indices of decision vector **/
          H_P2 = 0,
          H_P3 = 1,
          H_P4 = 2,
            /** Number of steps corresponds to number of state variable
                increments in scenario file **/
          steps = 11,
            /** Other variables **/
          i,j,k;

  DECIMAL
            /** Smallest non-zero proportion value **/
          base_p = 0.001,
            /** Exponent to linearize relationship between logit and
                proportion **/
          power = 2.5,
            /** Mean rates of hardwood encroachment **/
            /**  Rates set HIGH in this version **/
          e_base = 0.0006,
          e_P1 = 0.0012,
          e_P2 = 0.03,
          e_P3 = 0.03,
          e_P4 = 0.06,
            /** UH to P1 conversion fraction **/
            /**  Rate set LOW in this version **/
          q = 0.125,
            /** P3_RCW admission rate **/
            /**  Rate set LOW in this version **/
          p = 0.25,
            /** (inverse) age class transition rates **/
          t21 = 16.,
          t32 = 24.,
          t43 = 40.,
          t14 = 20.,   /** HIGH mortality rate in oldest age class **/
            /** Realized rates of hardwood encroachment **/
          e1, e2, e3, e4,
            /** Harvest-adjusted rates of hardwood encroachment **/
          f1, f2, f3, f4,
            /** Rates of harvest **/
          d2, d3, d4,
            /** Realized rate of catastrophic loss **/
          c,
            /** Catastrophic loss matrix **/
          C[6][6],
            /** Age class transition matrix **/
          G[6][6],
            /** Hardwood encroachment transition matrix **/
          H[6][6],
            /** Decision matrix **/
          D[6][6],
            /** Other temporary quantities **/
          P[6][6],
          T1[6][6],
          T2[6][6],
          this_f[6],
```

Appendix C.10. ASDP input files for forest model F1.

```
        next_f[6],
        half, scale, lrat, diff, denom, blogit,
        l1, l2, l3, l4, g1, g2, g3, g4;

/** HALF is the half-way increment **/
half = (steps+1.)/2.;
/** Logit value for smallest non-zero proportion value  (constant in
      denominator is number of components-1)                **/
blogit = log(base_p/(1.-4*base_p));
/** SCALE is the scale parameter for the polynomial curve (paired negation
      operators allow exponentiation to fractional power) **/
scale = -pow(-(1.-half),power)/blogit;

/** Read current state variables and convert them to logits
     (Paired negation operators allow exponentiation to fractional power) **/
diff = cur_state[P1-1]-half;
if (diff < 0) l1 = -pow(-diff,power)/scale;
         else l1 = pow(diff,power)/scale;
diff = cur_state[P2-1]-half;
if (diff < 0) l2 = -pow(-diff,power)/scale;
         else l2 = pow(diff,power)/scale;
diff = cur_state[P3-1]-half;
if (diff < 0) l3 = -pow(-diff,power)/scale;
         else l3 = pow(diff,power)/scale;
diff = cur_state[P4-1]-half;
if (diff < 0) l4 = -pow(-diff,power)/scale;
         else l4 = pow(diff,power)/scale;
denom = 1.+exp(l1)+exp(l2)+exp(l3)+exp(l4);

/** Calculate current forest composition states **/
this_f[RG] = 0;
this_f[P1] = exp(l1)/denom;
this_f[P2] = exp(l2)/denom;
this_f[P3] = exp(l3)/denom;
this_f[P4] = exp(l4)/denom;
this_f[UH] = 1./denom;

/** Set hardwood encroachment rates in proportion to rate for P1 **/
e1 = outcome[0] * e_P1/e_base;
e2 = outcome[0] * e_P2/e_base;
e3 = outcome[0] * e_P3/e_base;
e4 = outcome[0] * e_P4/e_base;
```

```
  /** Set catastrophic loss rate **/
  c = outcome[1];

  /** Matrix C (catastrophic loss) **/
  C[0][0] = 1.; C[0][1] = 0.; C[0][2] = 0.;   C[0][3] = 0.;   C[0][4] = 0.;   C[0][5]
= 0.;
  C[1][0] = 0.; C[1][1] = 1.; C[1][2] = c;    C[1][3] = c;    C[1][4] = c;    C[1][5]
= c*q;
  C[2][0] = 0.; C[2][1] = 0.; C[2][2] = 1.-c; C[2][3] = 0.;   C[2][4] = 0.;   C[2][5]
= 0.;
  C[3][0] = 0.; C[3][1] = 0.; C[3][2] = 0.;   C[3][3] = 1.-c; C[3][4] = 0.;   C[3][5]
= 0.;
  C[4][0] = 0.; C[4][1] = 0.; C[4][2] = 0.;   C[4][3] = 0.;   C[4][4] = 1.-c; C[4][5]
= 0.;
  C[5][0] = 0.; C[5][1] = 0.; C[5][2] = 0.;   C[5][3] = 0.;   C[5][4] = 0.;   C[5][5]
= 1.-c*q;

  /** Matrix G (age class transitions) **/
  g1 = 1./t21;
  g2 = 1./t32;
  g3 = 1./t43;
  g4 = 1./t14;

  G[0][0] = 0.; G[0][1] = 0.;     G[0][2] = 0.;     G[0][3] = 0.;         G[0][4] =
0.;     G[0][5] = 0.;
  G[1][0] = 1.; G[1][1] = 1.-g1;  G[1][2] = 0.;     G[1][3] = (1.-p)*g3;  G[1][4] =
g4;     G[1][5] = 0.;
  G[2][0] = 0.; G[2][1] = g1;     G[2][2] = 1.-g2;  G[2][3] = 0.;         G[2][4] =
0.;     G[2][5] = 0.;
  G[3][0] = 0.; G[3][1] = 0.;     G[3][2] = g2;     G[3][3] = 1.-g3;      G[3][4] =
0.;     G[3][5] = 0.;
  G[4][0] = 0.; G[4][1] = 0.;     G[4][2] = 0.;     G[4][3] = p*g3;       G[4][4] =
1.-g4;  G[4][5] = 0.;
  G[5][0] = 0.; G[5][1] = 0.;     G[5][2] = 0.;     G[5][3] = 0.;         G[5][4] =
0.;     G[5][5] = 1.;

  /** Matrix H (hardwood encroachment transitions) **/
  d2 = dec[H_P2];
  d3 = dec[H_P3];
  d4 = dec[H_P4];

  f1 = e1;
  f2 = max( 0, e2-d2);
```

Appendix C.10. ASDP input files for forest model F1.

```
  f3 = max( 0, e3-d3);
  f4 = max( 0, e4-d4);

  H[0][0] = 1.; H[0][1] = 0.;    H[0][2] = 0.;    H[0][3] = 0.;    H[0][4] = 0.;
H[0][5] = 0.;
  H[1][0] = 0.; H[1][1] = 1.-f1; H[1][2] = 0.;    H[1][3] = 0.;    H[1][4] = 0.;
H[1][5] = 0.;
  H[2][0] = 0.; H[2][1] = 0.;    H[2][2] = 1.-f2; H[2][3] = 0.;    H[2][4] = 0.;
H[2][5] = 0.;
  H[3][0] = 0.; H[3][1] = 0.;    H[3][2] = 0.;    H[3][3] = 1.-f3; H[3][4] = 0.;
H[3][5] = 0.;
  H[4][0] = 0.; H[4][1] = 0.;    H[4][2] = 0.;    H[4][3] = 0.;    H[4][4] = 1.-f4;
H[4][5] = 0.;
  H[5][0] = 0.; H[5][1] = f1;    H[5][2] = f2;    H[5][3] = f3;    H[5][4] = f4;
H[5][5] = 1.;

  /** Matrix D (harvest decisions) **/
  D[0][0] = 1.; D[0][1] = 0.; D[0][2] = d2;    D[0][3] = d3;    D[0][4] = d4;
D[0][5] = 0.;
  D[1][0] = 0.; D[1][1] = 1.; D[1][2] = 0.;    D[1][3] = 0.;    D[1][4] = 0.;
D[1][5] = 0.;
  D[2][0] = 0.; D[2][1] = 0.; D[2][2] = 1.-d2; D[2][3] = 0.;    D[2][4] = 0.;
D[2][5] = 0.;
  D[3][0] = 0.; D[3][1] = 0.; D[3][2] = 0.;    D[3][3] = 1.-d3; D[3][4] = 0.;
D[3][5] = 0.;
  D[4][0] = 0.; D[4][1] = 0.; D[4][2] = 0.;    D[4][3] = 0.;    D[4][4] = 1.-d4;
D[4][5] = 0.;
  D[5][0] = 0.; D[5][1] = 0.; D[5][2] = 0.;    D[5][3] = 0.;    D[5][4] = 0.;
D[5][5] = 1.;

  /** initialize NEXT_F and temporary matrices **/
  for (i=0; i<6; i=i+1)
  {
    next_f[i] = 0.;
    for (j=0; j<6; j=j+1)
    {
      T1[i][j] = 0.;
      T2[i][j] = 0.;
      P[i][j] = 0.;
    }
  }

  /** Compute (C*G) **/
```

```
  for (i=0; i<6; i=i+1)
    for (j=0; j<6; j=j+1)
      for (k=0; k<6; k=k+1)
        T1[i][j] = T1[i][j] + C[i][k]*G[k][j];

  /** Compute ((C*G) * H) **/
  for (i=0; i<6; i=i+1)
    for (j=0; j<6; j=j+1)
      for (k=0; k<6; k=k+1)
        T2[i][j] = T2[i][j] + T1[i][k]*H[k][j];

  /** Compute ( ((C*G) * H) ) * D) **/
  for (i=0; i<6; i=i+1)
    for (j=0; j<6; j=j+1)
      for (k=0; k<6; k=k+1)
        P[i][j] = P[i][j] + T2[i][k]*D[k][j];

  /** Advance to next forest composition state **/
  for (i=0; i<6; i=i+1)
    for (k=0; k<6; k=k+1)
      next_f[i] = next_f[i] + P[i][k]*this_f[k];


/*
printf("\nthis %f %f %f %f %f
%f",this_f[0],this_f[1],this_f[2],this_f[3],this_f[4],this_f[5]);
printf("\nnext %f %f %f %f %f
%f",next_f[0],next_f[1],next_f[2],next_f[3],next_f[4],next_f[5]);
printf("\ndecisions %f %f %f",d2,d3,d4);

printf("\n%f %f %f %f %f %f",G[0][0],G[0][1],G[0][2],G[0][3],G[0][4],G[0][5]);
printf("\n%f %f %f %f %f %f",G[1][0],G[1][1],G[1][2],G[1][3],G[1][4],G[1][5]);
printf("\n%f %f %f %f %f %f",G[2][0],G[2][1],G[2][2],G[2][3],G[2][4],G[2][5]);
printf("\n%f %f %f %f %f %f",G[3][0],G[3][1],G[3][2],G[3][3],G[3][4],G[3][5]);
printf("\n%f %f %f %f %f %f",G[4][0],G[4][1],G[4][2],G[4][3],G[4][4],G[4][5]);
printf("\n%f %f %f %f %f %f",G[5][0],G[5][1],G[5][2],G[5][3],G[5][4],G[5][5]);
*/

  /** Convert logit values to state values **/
/*printf("\ngot here P1 %f %f",next_f[P1],next_f[UH]);*/
  lrat = log(next_f[P1]/next_f[UH]);
  if (lrat<0) nxt_state[P1-1] = -pow((-lrat*scale),(1./power)) + half;
        else nxt_state[P1-1] = pow((lrat*scale),(1./power)) + half;
```

Appendix C.10. ASDP input files for forest model F1.

```
/*printf("\ngot here P2 %f %f",next_f[P2],next_f[UH]);*/
  lrat = log(next_f[P2]/next_f[UH]);
  if (lrat<0) nxt_state[P2-1] = -pow((-lrat*scale),(1./power)) + half;
         else nxt_state[P2-1] = pow((lrat*scale),(1./power)) + half;
/*printf("\ngot here P3 %f %f",next_f[P3],next_f[UH]);*/
  lrat = log(next_f[P3]/next_f[UH]);
  if (lrat<0) nxt_state[P3-1] = -pow((-lrat*scale),(1./power)) + half;
         else nxt_state[P3-1] = pow((lrat*scale),(1./power)) + half;
/*printf("\ngot here P4 %f %f",next_f[P4],next_f[UH]);*/
  lrat = log(next_f[P4]/next_f[UH]);
  if (lrat<0) nxt_state[P4-1] = -pow((-lrat*scale),(1./power)) + half;
         else nxt_state[P4-1] = pow((lrat*scale),(1./power)) + half;

  return;
}
```

Stage Return File

```
#include "obj.h"
/***********************************************************************
 PNWR forest succession model
 Stage return calculation
 version HWD1 -- includes hardwood component (UH) w/encroachment rates,
                 logit expression of state space
   STATE has 4 elements:  P1, P2, P3, P3RCW
     --> 5th state is UH which is obtained by subtraction <--
**********************************************************************/
{
  INT
            /** Indices of state vector **/
          P1 = 0,
          P2 = 1,
          P3 = 2,
          P4 = 3,
          UH = 4,
            /** Number of steps corresponds to number of state variable
                increments in scenario file **/
          steps = 11;
DECIMAL
          /** Smallest non-zero proportion value **/
        base_p = 0.001,
          /** Exponent to linearize relationship between logit and
              proportion **/
        power = 2.5,
        P3R_threshold = 0.030961749,
          /** Other temporary quantities **/
        f[5],
        half, scale, diff, l1, l2, l3, l4, denom, blogit;

/** HALF is the half-way increment **/
half = (steps+1.)/2.;
/** Logit value for smallest non-zero proportion value  (constant in
      denominator is number of components-1)                **/
blogit = log(base_p/(1.-4*base_p));
/** SCALE is the scale parameter for the polynomial curve (paired negation
      operators allow exponentiation to fractional power) **/
scale = -pow(-(1.-half),power)/blogit;

/** Read current state variables and convert them to logits
     (Paired negation operators allow exponentiation to fractional power) **/
diff = cur_state[P1]-half;
if (diff < 0) l1 = -pow(-diff,power)/scale;
         else l1 = pow(diff,power)/scale;
diff = cur_state[P2]-half;
if (diff < 0) l2 = -pow(-diff,power)/scale;
         else l2 = pow(diff,power)/scale;
diff = cur_state[P3]-half;
if (diff < 0) l3 = -pow(-diff,power)/scale;
         else l3 = pow(diff,power)/scale;
diff = cur_state[P4]-half;
if (diff < 0) l4 = -pow(-diff,power)/scale;
         else l4 = pow(diff,power)/scale;
denom = 1.+exp(l1)+exp(l2)+exp(l3)+exp(l4);

/** Calculate current forest composition states **/
f[P1] = exp(l1)/denom;
f[P2] = exp(l2)/denom;
f[P3] = exp(l3)/denom;
f[P4] = exp(l4)/denom;
f[UH] = 1./denom;
```

Appendix C.10. ASDP input files for forest model F1.

```
  /** Amount of P3_RCW (P4) habitat must exceed threshold value.  If it does,
        return is 1000+(habitat-threshold).  If it does not, return is 0. **/
  if (f[P4]<P3R_threshold) result = 0.;
  else result = 1000. + (f[P4]-threshold);

 return  (result);
}
```

Scenario File

```
!-------------------------------------------------------------------
! PNWR forest succession model -- version HWD1, ALT1
!-------------------------------------------------------------------
MAX;
TITLE   ">>> PNWR forest succession model (HWD1) <<<"   ;
!REPORT_ALL      ;
!TRACE ;
ITERATIONS   20|500;
!ITERATIONS 1;
NO_CHANGE 3;

! Simulation settings
! Initial forest state is (P1=0.1227 P2=0.0741 P3=0.6173 P3R=0.001 UH=0.1849)
INIT_STATE  4.3838546  3.7726398  8.4877613  1.5290676;
SIM_TRIALS 1000;
SIM_REPORT 1 1 1 1 1 1 1;
NO_INTERP;

STATE   "P1 index";
STATE   "P2 index";
STATE   "P3 index";
STATE   "P3R index";

DECISION "P2 harvest";
DECISION "P3 harvest";
DECISION "P3R harvest";

DISTRIB "Hdwd P1";
RV "Hdwd P1";
!EVENT 1.0 0.0006;

EVENT      0.1    0.000072137044;
EVENT      0.2     0.00020748589;
EVENT      0.4     0.00048900905;
EVENT      0.2     0.00094358994;
EVENT      0.1      0.0016696751;

DISTRIB "Cat rate";
RV "Cat rate";
!EVENT 1.0 0.003394;

EVENT   0.500   0.00000000000000000000000093853655;
EVENT   0.400   0.0000014526682;
EVENT   0.050   0.00052868322;
EVENT   0.045   0.019829935;
EVENT   0.005   0.15752754;

STAGE 1 1 1 1;
COMBINE 1|11 1|11 1|11 1|11,
        0|0.14|0.02 0|0.14|0.02 0|0.14|0.02,
        "Hdwd P1" "Cat rate";
```

Appendix C.11. ASDP input files for forest model F2. State dynamics, stage return, and scenario files corresponding to model of slow rate of hardwood encroachment.

State Dynamics File

```
#include "sd.h"
/*********************************************************************
 PNWR forest succession model (sensitivity analysis version ALT2)
 version HWD1 -- includes hardwood component (UH) w/encroachment rates,
                 logit expression of state space
   CUR_STATE has 4 elements:  P1, P2, P3, P3RCW
     --> 5th state is UH which is obtained by subtraction <--
     A temporary 6th state (RG) is used to compute additions to the P1 class.
       It is emptied during the transitions.  Because of this additional
       component, the relationship between the temporary vectors and the
       DP vectors (CUR_STATE and NXT_STATE) is
                        CUR_STATE[i] = THIS_F[i+1].
   DEC has 3 elements:  P2 harvest, P3 harvest, P3RCW harvest
   OUTCOME has 2 elements:  overall encroachment and catastrophic loss rates
********************************************************************/
{
  INT
            /** Indices of temporary state vectors **/
          RG = 0,
          P1 = 1,
          P2 = 2,
          P3 = 3,
          P4 = 4,
          UH = 5,
            /** Indices of decision vector **/
          H_P2 = 0,
          H_P3 = 1,
          H_P4 = 2,
            /** Number of steps corresponds to number of state variable
                increments in scenario file **/
          steps = 11,
            /** Other variables **/
          i,j,k;

  DECIMAL
            /** Smallest non-zero proportion value **/
          base_p = 0.001,
            /** Exponent to linearize relationship between logit and
      proportion **/
power = 2.5,
  /** Mean rates of hardwood encroachment **/
  /**  Rates set LOW in this version **/
e_base = 0.0006,
e_P1 = 0.0003,
e_P2 = 0.0075,
e_P3 = 0.0075,
e_P4 = 0.015,
  /** UH to P1 conversion fraction **/
  /**  Rate set HIGH in this version **/
q = 0.5,
  /** P3_RCW admission rate **/
  /**  Rate set HIGH in this version **/
p = 1.0,
  /** (inverse) age class transition rates **/
t21 = 16.,
t32 = 24.,
t43 = 40.,
t14 = 60.,   /** LOW mortality rate in oldest age class **/
  /** Realized rates of hardwood encroachment **/
e1, e2, e3, e4,
  /** Harvest-adjusted rates of hardwood encroachment **/
f1, f2, f3, f4,
  /** Rates of harvest **/
d2, d3, d4,
  /** Realized rate of catastrophic loss **/
c,
  /** Catastrophic loss matrix **/
C[6][6],
  /** Age class transition matrix **/
G[6][6],
  /** Hardwood encroachment transition matrix **/
H[6][6],
  /** Decision matrix **/
D[6][6],
  /** Other temporary quantities **/
P[6][6],
T1[6][6],
T2[6][6],
this_f[6],
```

Appendix C.11. ASDP input files for forest model F2.

```
        next_f[6],
        half, scale, lrat, diff, denom, blogit,
        l1, l2, l3, l4, g1, g2, g3, g4;

/** HALF is the half-way increment **/
half = (steps+1.)/2.;
/** Logit value for smallest non-zero proportion value  (constant in
      denominator is number of components-1)                **/
blogit = log(base_p/(1.-4*base_p));
/** SCALE is the scale parameter for the polynomial curve (paired negation
      operators allow exponentiation to fractional power) **/
scale = -pow(-(1.-half),power)/blogit;

/** Read current state variables and convert them to logits
     (Paired negation operators allow exponentiation to fractional power) **/
diff = cur_state[P1-1]-half;
if (diff < 0) l1 = -pow(-diff,power)/scale;
         else l1 = pow(diff,power)/scale;
diff = cur_state[P2-1]-half;
if (diff < 0) l2 = -pow(-diff,power)/scale;
         else l2 = pow(diff,power)/scale;
diff = cur_state[P3-1]-half;
if (diff < 0) l3 = -pow(-diff,power)/scale;
         else l3 = pow(diff,power)/scale;
diff = cur_state[P4-1]-half;
if (diff < 0) l4 = -pow(-diff,power)/scale;
         else l4 = pow(diff,power)/scale;
denom = 1.+exp(l1)+exp(l2)+exp(l3)+exp(l4);

/** Calculate current forest composition states **/
this_f[RG] = 0;
this_f[P1] = exp(l1)/denom;
this_f[P2] = exp(l2)/denom;
this_f[P3] = exp(l3)/denom;
this_f[P4] = exp(l4)/denom;
this_f[UH] = 1./denom;

/** Set hardwood encroachment rates in proportion to rate for P1 **/
e1 = outcome[0] * e_P1/e_base;
e2 = outcome[0] * e_P2/e_base;
e3 = outcome[0] * e_P3/e_base;
e4 = outcome[0] * e_P4/e_base;
```

```
  /** Set catastrophic loss rate **/
  c = outcome[1];

  /** Matrix C (catastrophic loss) **/
  C[0][0] = 1.; C[0][1] = 0.; C[0][2] = 0.;   C[0][3] = 0.;   C[0][4] = 0.;   C[0][5]
= 0.;
  C[1][0] = 0.; C[1][1] = 1.; C[1][2] = c;    C[1][3] = c;    C[1][4] = c;    C[1][5]
= c*q;
  C[2][0] = 0.; C[2][1] = 0.; C[2][2] = 1.-c; C[2][3] = 0.;   C[2][4] = 0.;   C[2][5]
= 0.;
  C[3][0] = 0.; C[3][1] = 0.; C[3][2] = 0.;   C[3][3] = 1.-c; C[3][4] = 0.;   C[3][5]
= 0.;
  C[4][0] = 0.; C[4][1] = 0.; C[4][2] = 0.;   C[4][3] = 0.;   C[4][4] = 1.-c; C[4][5]
= 0.;
  C[5][0] = 0.; C[5][1] = 0.; C[5][2] = 0.;   C[5][3] = 0.;   C[5][4] = 0.;   C[5][5]
= 1.-c*q;

  /** Matrix G (age class transitions) **/
  g1 = 1./t21;
  g2 = 1./t32;
  g3 = 1./t43;
  g4 = 1./t14;

  G[0][0] = 0.; G[0][1] = 0.;     G[0][2] = 0.;     G[0][3] = 0.;         G[0][4] =
0.;     G[0][5] = 0.;
  G[1][0] = 1.; G[1][1] = 1.-g1;  G[1][2] = 0.;     G[1][3] = (1.-p)*g3;  G[1][4] =
g4;     G[1][5] = 0.;
  G[2][0] = 0.; G[2][1] = g1;     G[2][2] = 1.-g2;  G[2][3] = 0.;         G[2][4] =
0.;     G[2][5] = 0.;
  G[3][0] = 0.; G[3][1] = 0.;     G[3][2] = g2;     G[3][3] = 1.-g3;      G[3][4] =
0.;     G[3][5] = 0.;
  G[4][0] = 0.; G[4][1] = 0.;     G[4][2] = 0.;     G[4][3] = p*g3;       G[4][4] =
1.-g4;  G[4][5] = 0.;
  G[5][0] = 0.; G[5][1] = 0.;     G[5][2] = 0.;     G[5][3] = 0.;         G[5][4] =
0.;     G[5][5] = 1.;

  /** Matrix H (hardwood encroachment transitions) **/
  d2 = dec[H_P2];
  d3 = dec[H_P3];
  d4 = dec[H_P4];

  f1 = e1;
  f2 = max( 0, e2-d2);
```

Appendix C.11. ASDP input files for forest model F2.

```
  f3 = max( 0, e3-d3);
  f4 = max( 0, e4-d4);

  H[0][0] = 1.; H[0][1] = 0.;    H[0][2] = 0.;    H[0][3] = 0.;    H[0][4] = 0.;
H[0][5] = 0.;
  H[1][0] = 0.; H[1][1] = 1.-f1; H[1][2] = 0.;    H[1][3] = 0.;    H[1][4] = 0.;
H[1][5] = 0.;
  H[2][0] = 0.; H[2][1] = 0.;    H[2][2] = 1.-f2; H[2][3] = 0.;    H[2][4] = 0.;
H[2][5] = 0.;
  H[3][0] = 0.; H[3][1] = 0.;    H[3][2] = 0.;    H[3][3] = 1.-f3; H[3][4] = 0.;
H[3][5] = 0.;
  H[4][0] = 0.; H[4][1] = 0.;    H[4][2] = 0.;    H[4][3] = 0.;    H[4][4] = 1.-f4;
H[4][5] = 0.;
  H[5][0] = 0.; H[5][1] = f1;    H[5][2] = f2;    H[5][3] = f3;    H[5][4] = f4;
H[5][5] = 1.;

  /** Matrix D (harvest decisions) **/
  D[0][0] = 1.; D[0][1] = 0.; D[0][2] = d2;    D[0][3] = d3;    D[0][4] = d4;
D[0][5] = 0.;
  D[1][0] = 0.; D[1][1] = 1.; D[1][2] = 0.;    D[1][3] = 0.;    D[1][4] = 0.;
D[1][5] = 0.;
  D[2][0] = 0.; D[2][1] = 0.; D[2][2] = 1.-d2; D[2][3] = 0.;    D[2][4] = 0.;
D[2][5] = 0.;
  D[3][0] = 0.; D[3][1] = 0.; D[3][2] = 0.;    D[3][3] = 1.-d3; D[3][4] = 0.;
D[3][5] = 0.;
  D[4][0] = 0.; D[4][1] = 0.; D[4][2] = 0.;    D[4][3] = 0.;    D[4][4] = 1.-d4;
D[4][5] = 0.;
  D[5][0] = 0.; D[5][1] = 0.; D[5][2] = 0.;    D[5][3] = 0.;    D[5][4] = 0.;
D[5][5] = 1.;

  /** initialize NEXT_F and temporary matrices **/
  for (i=0; i<6; i=i+1)
  {
    next_f[i] = 0.;
    for (j=0; j<6; j=j+1)
    {
      T1[i][j] = 0.;
      T2[i][j] = 0.;
      P[i][j] = 0.;
    }
  }

  /** Compute (C*G) **/
```

```
  for (i=0; i<6; i=i+1)
    for (j=0; j<6; j=j+1)
      for (k=0; k<6; k=k+1)
        T1[i][j] = T1[i][j] + C[i][k]*G[k][j];

  /** Compute ((C*G) * H) **/
  for (i=0; i<6; i=i+1)
    for (j=0; j<6; j=j+1)
      for (k=0; k<6; k=k+1)
        T2[i][j] = T2[i][j] + T1[i][k]*H[k][j];

  /** Compute ( ((C*G) * H) ) * D) **/
  for (i=0; i<6; i=i+1)
    for (j=0; j<6; j=j+1)
      for (k=0; k<6; k=k+1)
        P[i][j] = P[i][j] + T2[i][k]*D[k][j];

  /** Advance to next forest composition state **/
  for (i=0; i<6; i=i+1)
    for (k=0; k<6; k=k+1)
      next_f[i] = next_f[i] + P[i][k]*this_f[k];


/*
printf("\nthis %f %f %f %f %f
%f",this_f[0],this_f[1],this_f[2],this_f[3],this_f[4],this_f[5]);
printf("\nnext %f %f %f %f %f
%f",next_f[0],next_f[1],next_f[2],next_f[3],next_f[4],next_f[5]);
printf("\ndecisions %f %f %f",d2,d3,d4);

printf("\n%f %f %f %f %f %f",G[0][0],G[0][1],G[0][2],G[0][3],G[0][4],G[0][5]);
printf("\n%f %f %f %f %f %f",G[1][0],G[1][1],G[1][2],G[1][3],G[1][4],G[1][5]);
printf("\n%f %f %f %f %f %f",G[2][0],G[2][1],G[2][2],G[2][3],G[2][4],G[2][5]);
printf("\n%f %f %f %f %f %f",G[3][0],G[3][1],G[3][2],G[3][3],G[3][4],G[3][5]);
printf("\n%f %f %f %f %f %f",G[4][0],G[4][1],G[4][2],G[4][3],G[4][4],G[4][5]);
printf("\n%f %f %f %f %f %f",G[5][0],G[5][1],G[5][2],G[5][3],G[5][4],G[5][5]);
*/

  /** Convert logit values to state values **/
/*printf("\ngot here P1 %f %f",next_f[P1],next_f[UH]);*/
  lrat = log(next_f[P1]/next_f[UH]);
  if (lrat<0) nxt_state[P1-1] = -pow((-lrat*scale),(1./power)) + half;
         else nxt_state[P1-1] = pow((lrat*scale),(1./power)) + half;
```

Appendix C.11. ASDP input files for forest model F2.

```
/*printf("\ngot here P2 %f %f",next_f[P2],next_f[UH]);*/
  lrat = log(next_f[P2]/next_f[UH]);
  if (lrat<0) nxt_state[P2-1] = -pow((-lrat*scale),(1./power)) + half;
         else nxt_state[P2-1] = pow((lrat*scale),(1./power)) + half;
/*printf("\ngot here P3 %f %f",next_f[P3],next_f[UH]);*/
  lrat = log(next_f[P3]/next_f[UH]);
  if (lrat<0) nxt_state[P3-1] = -pow((-lrat*scale),(1./power)) + half;
         else nxt_state[P3-1] = pow((lrat*scale),(1./power)) + half;
/*printf("\ngot here P4 %f %f",next_f[P4],next_f[UH]);*/
  lrat = log(next_f[P4]/next_f[UH]);
  if (lrat<0) nxt_state[P4-1] = -pow((-lrat*scale),(1./power)) + half;
         else nxt_state[P4-1] = pow((lrat*scale),(1./power)) + half;

  return;
}
```

Stage Return File

```
#include "obj.h"
/**********************************************************************
 PNWR forest succession model
 Stage return calculation
 version HWD1 -- includes hardwood component (UH) w/encroachment rates,
                 logit expression of state space
   STATE has 4 elements:  P1, P2, P3, P3RCW
     --> 5th state is UH which is obtained by subtraction <--
**********************************************************************/
{
  INT
            /** Indices of state vector **/
          P1 = 0,
          P2 = 1,
          P3 = 2,
          P4 = 3,
          UH = 4,
            /** Number of steps corresponds to number of state variable
                increments in scenario file **/
          steps = 11;
DECIMAL
          /** Smallest non-zero proportion value **/
        base_p = 0.001,
          /** Exponent to linearize relationship between logit and
              proportion **/
        power = 2.5,
        P3R_threshold = 0.030961749,
          /** Other temporary quantities **/
        f[5],
        half, scale, diff, l1, l2, l3, l4, denom, blogit;

/** HALF is the half-way increment **/
half = (steps+1.)/2.;
/** Logit value for smallest non-zero proportion value  (constant in
      denominator is number of components-1)                **/
blogit = log(base_p/(1.-4*base_p));
/** SCALE is the scale parameter for the polynomial curve (paired negation
      operators allow exponentiation to fractional power) **/
scale = -pow(-(1.-half),power)/blogit;

/** Read current state variables and convert them to logits
     (Paired negation operators allow exponentiation to fractional power) **/
diff = cur_state[P1]-half;
if (diff < 0) l1 = -pow(-diff,power)/scale;
         else l1 = pow(diff,power)/scale;
diff = cur_state[P2]-half;
if (diff < 0) l2 = -pow(-diff,power)/scale;
         else l2 = pow(diff,power)/scale;
diff = cur_state[P3]-half;
if (diff < 0) l3 = -pow(-diff,power)/scale;
         else l3 = pow(diff,power)/scale;
diff = cur_state[P4]-half;
if (diff < 0) l4 = -pow(-diff,power)/scale;
         else l4 = pow(diff,power)/scale;
denom = 1.+exp(l1)+exp(l2)+exp(l3)+exp(l4);

/** Calculate current forest composition states **/
f[P1] = exp(l1)/denom;
f[P2] = exp(l2)/denom;
f[P3] = exp(l3)/denom;
f[P4] = exp(l4)/denom;
f[UH] = 1./denom;
```

Appendix C.11. ASDP input files for forest model F2.

```
  /** Amount of P3_RCW (P4) habitat must exceed threshold value.  If it does,
        return is 1000+(habitat-threshold).  If it does not, return is 0. **/
  if (f[P4]<P3R_threshold) result = 0.;
  else result = 1000. + (f[P4]-threshold);

 return  (result);
}
```

Scenario File

```
!-------------------------------------------------------------------
! PNWR forest succession model -- version HWD1, ALT2
!-------------------------------------------------------------------
MAX;
TITLE   ">>> PNWR forest succession model (HWD1) <<<"   ;
!REPORT_ALL      ;
!TRACE ;
ITERATIONS   20|500;
!ITERATIONS 1;
NO_CHANGE 3;

! Simulation settings
! Initial forest state is (P1=0.1227 P2=0.0741 P3=0.6173 P3R=0.001 UH=0.1849)
INIT_STATE  4.3838546  3.7726398  8.4877613  1.5290676;
SIM_TRIALS 1000;
SIM_REPORT 1 1 1 1 1 1 1;
NO_INTERP;

STATE   "P1 index";
STATE   "P2 index";
STATE   "P3 index";
STATE   "P3R index";

DECISION "P2 harvest";
DECISION "P3 harvest";
DECISION "P3R harvest";

DISTRIB "Hdwd P1";
RV "Hdwd P1";
!EVENT 1.0 0.0006;

EVENT      0.1    0.00040810778;
EVENT      0.2    0.00049569219;
EVENT      0.4    0.00059272935;
EVENT      0.2    0.00070092313;
EVENT      0.1    0.00082774418;

DISTRIB "Cat rate";
RV "Cat rate";
!EVENT 1.0 0.003394;

EVENT   0.500   0.000000025733770;
EVENT   0.400   0.0015218462;
EVENT   0.050   0.023279722;
EVENT   0.045   0.081601660;
EVENT   0.005   0.25323891;

STAGE 1 1 1 1;
COMBINE 1|11 1|11 1|11 1|11,
        0|0.14|0.02 0|0.14|0.02 0|0.14|0.02,
        "Hdwd P1" "Cat rate";
```

Appendix C.12. Spatially-explicit Refuge simulation model. *PNWR_SIM.G* is the main GAUSS program file for simulating management on the Piedmont National Wildlife Refuge. All subroutine and data files are called by this program, and all user-controlled variables are specified in this file. Simulation results are saved in the GAUSS matrix file OUTCOMES. *PROCS1.G* contains mostly general-use mathematical and transformation procedures, and *PROCS2.G* contains procedures specific to the simulation model.

PNWR_SIM.G

```
@ Simulates burning and cutting management on the PNWR.  Inputs are (1) files
    of hexagon map attributes, (2) file of optimal regeneration decisions for
    lookup, (3) list of compartments to burn through time, and (4) list of
    compartments to cut through time.  Outputs are (1) distribution and
    number of hexagons occupied by RCWs and (2) densities of WOTH.  @

new;
library pgraph;

#include "d:\\pnwr\\procs1.g";

begtime = date;

rndseed 492407754;

@ Input path for hexagon attribute data @
pathname = "d:\\pnwr";

@ Input path for forest regeneration decision matrix @
dec_path = "d:\\pnwr\\";

@ File name of forest regeneration decision matrix @
dec_file = "hwd1_opt";

@ Parameters @
  start_yr = 2000;                     @ Start year @
  timemax = 100;                       @ Number of iterations @
  allcomp = 34;                        @ Total number of compartments @
  maxcomp = {5, 4, 4, 4, 4, 4, 4, 5};  @ Sizes of management groups @
  startgrp = 5;              @ Management group to start in year START_YR+1 @

  reps = 20;                 @ Number of decision replicates @
  strings = 9;               @ Number of decision strings @
  burn_max = {7, 17};        @ Max number of compartments to burn per year @
    @ BURN_MAX = 17 ==> 2-yr avg frequency @
    @           11 ==> 3-yr               @
    @            9 ==> 4-yr               @
    @            7 ==> 5-yr               @

  prodmod = 1;                @ RCW productivity model selection @
  wothmod = 1;                @ WOTH model selection @
  outpath = "d:\\pnwr";

@ Forest growth model parameters @
  thinindx = 0.6;     @ Values <1.0 represent degree of thinning from below @
  b1 = 2.81706;       @                                @
  b2 = -11935.2;      @ Bailey-Ware model coefficients @
  b3 = 0.043493;      @                                @
  age0 = 8;           @ Age at which BA follows Bailey-Ware model @
  ba0 = 0.46;         @ Constant BA (2.00 ft^2/ac) value up through age AGE0 @
  ba_max = 35;        @ Maximum BA (152 ft^2/ac) allowed.  BA model is       @
                      @ sensitive to errors introduced by discretization of   @
                      @ stands and treatment years into hexagons, especially  @
                      @ for P2 stands.                                        @
  @  m_rates = {0, 0.0125, 0.025}; @
  m_rates = {0, 0, 0};    @ Stand mortality rates for P2, P3, P4 @

@ Parameter for canopy closure model @
  ccrate = 0.79;

@ Recruitment cluster establishment parameters @
  clusgrow = 0.10;    @ Cluster growth rate @
  strmdist = 125;     @ Minimum stream distance @
  @ Nesting habitat requirements @
  rc_age1 = 40;
  rc_amt1 = 0.85;     @ Min quantity of RC_AGE1 or older @
  rc_age2 = 60;
  rc_amt2 = 0.50;     @ Min quantity of RC_AGE2 or older @
  @ 800-m (48-ring) foraging habitat requirements @
```

Appendix C.12. Spatially-explicit Refuge simulation model. Continued.

```
  fh1_age = 40;        @ min age @
  fh1_ba1 = 9.1827;    @ min BA (9.1827 m^2/ha=40 ft^2/ac) @
  fh1_ba2 = 18.3655;   @ max BA (18.3655 m^2/ha=80 ft^2/ac) @
  fh1_area = 505857;   @ min area of this habitat (505857 m^2=125 ac) @
  out800 = {0.5,0.2,0.2};    @ Estimated habitat provisions
                               for ONF, Priv-in, Priv-out @
  @ 400-m (12-ring) foraging habitat requirements @
  fh2_age = 40;        @ min age @
  fh2_ba1 = 9.1827;    @ min BA (9.1827 m^2/ha=40 ft^2/ac) @
  fh2_ba2 = 18.3655;   @ max BA (18.3655 m^2/ha=80 ft^2/ac) @
  fh2_area = 252929;   @ min area of this habitat (252929 m^2=62.5 ac) @
  out400 = {0.5,0.2,0.2};    @ Estimated habitat provisions
                               for ONF, Priv-in, Priv-out @
  @ Min nearest-neighbor distance to any cluster @
  clus_d1 = 402.336;     @ (402.336 m = 1/4 mile) @
  @ Max desired distance from any active cluster @
  clus_d2a = 1609.344;   @ (1609.344 m = 1 mile) @
  @ Max permissible distance from any active cluster @
  clus_d2b = 3218.688;   @ (3218.688 m = 2 miles) @
  @ Active cluster distance increment @
  clus_inc = 402.336;    @ (402.336 m = 1/4 mile) @

@ Parameters for transformations used in regeneration decision look-up @
  dec_parm = {11, 0.001, 2.5};    @ Parameters STEPS, BASE_P, and POWER @

@ Weights for geometric avg of area and distance for regeneration site choice @
  wt_area = 1;
  wt_dist = 1;

@ Regeneration patch size limit @
  cutlimit = 101171;     @ (101171 m^2 = 25 ac) @

@ Understory vegetation model parameters @
  v1 = -1461.4;
  v2 = 36554.6;
  v3 = 21.4;
  v4 = 100;
  v5 = -14.8;
  v6 = 13.8;

@ Vegetation reductions following burns @
  burnparm = {0.76, 0.87, 0.97};    @ Years 1, 2, 3, post-burn, respectively @
```

```
@ Thinning parameters @
  ba_thin = 13.7741;   @ residual BA (13.7741 m^2/ha=60 ft^2/ac) @
  ba_ceil = 18.3655;   @ ceiling BA (18.3655 m^2/ha=80 ft^2/ac) @
  thin_age = 16;       @ min age eligible for thinning @

@ Hardwood encroachment parameters @
  burnlim = 5;    @ Years since burning required before encroachment occurs @
  thinlim = 5;    @ Years since thinning required before encroachment occurs @
  hwd_p1 = 0.0006;  @ Annual average rate of P1 pine loss to hardwood @
  hwd_p2 = 0.015;   @ Annual average rate of P2 pine loss to hardwood @
  hwd_p3 = 0.015;   @ Annual average rate of P3 pine loss to hardwood @
  hwd_p4 = 0.03;    @ Annual average rate of P4 pine loss to hardwood @
  hwd_cv = 40;      @ CV (percent) for random encroachment rate @

  hwd_parm = hwd_p1 | hwd_p2 | hwd_p3 | hwd_p4;
  if hwd_cv<=0;
    hwd_beta = 0;
  else;
    hwd_parm = hwd_parm ~ (hwd_parm*hwd_cv/100)^2;
    hwd_beta = betaparm(hwd_parm[1,.],1);
  endif;

@ Forest disturbance parameters @

  dstrb_mn = 0.003394;    @ Mean rate (proportion) of disturbance @
  dstrb_vr = 0.0003572;   @ Variance of disturbance proportion @
  dstrb_hw = 0.25;     @ Proportion of disturbed UH area that regens as pine @

  if dstrb_vr<=0;
    dstrb_b = 0;
  else;
    dstrb_b = betaparm(dstrb_mn~dstrb_vr,1);
  endif;

@ Logistic regression parameters from Loeb et al study @
  loebparm = {5.134, -0.45738};
  ba_adj = 1 ~ (0/4.356);  @ Scale and shift parameters for alt Loeb model @
  ba_adj = 1.05 ~ (10/4.356);  @ Scale and shift parameters for alt Loeb model @
  ba_adj = 1.02 ~ (5/4.356);  @ Scale and shift parameters for alt Loeb model @
  ba_adj = 1.2 ~ (30/4.356);  @ Scale and shift parameters for alt Loeb model @

@ Understory density-midstory BA conversion parameters @
  tba = 2/3;             @ Exponentiation applied to veg density @
```

Appendix C.12. Spatially-explicit Refuge simulation model. Continued.

```
  avg_tvd = 85.132063;   @ Mean transformed veg dens (kg/ha) in >80 yr stands@
  avg_mba = 39.962264;   @ Weighted mean of midstory BA (ft^2/ac) from Loeb @
  midba = tba | avg_tvd | avg_mba;

@ RCW productivity parameters @
  @  Distribution of nestling group sizes @
  nestling = {0 0.405356954550248,
              1 0.114837076874407,
              2 0.263699954304194,
              3 0.191641182466871,
              4 0.0244648318042813};
  mn_nestl = nestling[.,1]'*nestling[.,2];  @ Mean nestling group size @
  fa_0 =  75;          @ 'Intercept' value for psi function @
  surv_fl = 0.380;     @ Fledgling survival value from Maguire et al @
  model1 = {0, 0};     @ Model 1 values for theta, tau @
  model2 = {0.8, 2};   @ Model 2 values for theta, tau @
  prodparm = model1 ~ model2;
  logit_fl = log(surv_fl/(1-surv_fl));
  expon = (nestling[.,2]/mn_nestl - 1)';

@ RCW recruitment probability parameters @
  recrparm = {0.080979, -0.69077};  @ I_0 and B2, respectively @

  @ "Average" X-Y location and productivity in HEF @
  hef_x = 250500;
  hef_y = 3661000;
  hef_prod = 6;

@ WOTH model parameters @
  @  Intercept, time since burning parameters for low veg model @
  woth_p1 = {0.34823 0.02332};
  @  Intercept, time since burning, veg density parms for veg logit model @
  woth_p2 = {-0.59658 0.03339 0.00015544};

  @ Veg input limits for WOTH models @
  ba_lims = {47.1429, 80};
  cc_lims = {0.739865, 0.954545};
  low_lims = {0.310667, 0.753401};
  log_lims = {-0.681402, 0.343801};

  @  Intercept, BA, CC, low veg, BA*CC parameters for hardwood WOTH model @
  woth_p3 = { 18.89173 -0.45437 -26.77392 -1.82827 0.55391 };
  @  Variance-covariance matrix for WOTH_P3 @
  woth_p3v = { 501.33068175    -7.423478 -556.9614474  -4.89917432 8.2981171871,
                  -7.423478 0.1116236093 8.2514513481 0.0862719842 -0.124978159,
               -556.9614474 8.2514513481 620.60796105  5.045491335 -9.247523585,
                -4.89917432 0.0862719842  5.045491335 1.7273684569 -0.105250231,
               8.2981171871 -0.124978159 -9.247523585 -0.105250231 0.1403682674};

  @ Intercept, BA, low veg, veg logit, BA*logit, low*logit for pine WOTH model@
  woth_p4 = { -7.80291 0.09640 -4.65806 -14.80672 0.25183 -5.99684 };
  @  Variance-covariance matrix for WOTH_P4 @
  woth_p4v =
  {2.6595292625 -0.043487868 0.5536141562  7.002288716 -0.117015531 1.7837461772,
   -0.043487868 0.0007602069 -0.015342165  -0.11705057 0.0020299843 -0.039635776,
   0.5536141562 -0.015342165 0.9650422103 1.7847983221 -0.039930765 1.8845440159,
    7.002288716  -0.11705057 1.7847983221 22.135985069 -0.359324972 3.9725262098,
   -0.117015531 0.0020299843 -0.039930765 -0.359324972 0.0062131852 -0.116866731,
   1.7837461772 -0.039635776 1.8845440159 3.9725262098 -0.116866731 8.2933866172 };


@ Columns of HEX_FOR input matrix:
  1     2   3  4 5    6    7    8        9      10       11     12    13
hex_id row col x y inside c1 trtyear prevyear lastburn lastthin d_hwb d_hwu
 14  15   16   17   18  19   20   21   22  23   24      25
d_o d_w area1 age1 ba1 cc1 area2 age2 ba2 cc2 rcw2000 clustype
@

@ HEX_FOR is the data matrix of hexagon indices, treatment variables, and
    forest cover attributes.  HDR_FOR is a matrix of field names and width
    and precision information @
load path=^pathname hex_for, hdr_for;

@ Columns of HEX_ATTR input matrix:
   1      2      3      4       5       6     7     8      9
hex_id i_medn p_medn s_dens d_stream d_pnwr d_onf d_p_in d_p_out
    10     11     12      13     14     15     16     17
b4_pnwr b4_onf b4_pin b4_pout b8_pnwr b8_onf b8_pin b8_pout
    18       19      20      21       22       23      24      25
b16_pnwr b16_onf b16_pin b16_pout b32_pnwr b32_onf b32_pin b32_pout
@

@ HEX_ATTR is the data matrix of hexagon physical attributes, such as
   site index, slope position, distances, and ownership proportions.
   HDR_ATTR is a maxtrix of attribute names and precision information @
load path=^pathname hex_attr, hdr_attr;
```

```
@ Load data from forest regeneration decision matrix @
infile = dec_path $+ dec_file;
open fh = ^infile for read;
dectable = readr(fh,15000);
fh = close(fh);

@ Values for rectangular grid dimensions from SAS grid creation program @
gridsize = ( 1 ~ 1 ) | ( 101 ~ 98 );

@ Full hexagon size @;
fullsize = maxc(hex_for[.,6]);

@ Rows in F_STATE matrix @;
hexes = rows(hex_for);

@ Compute and save inter-cell distances
    After computing the first time, comment out following 2 lines and
    use LOAD statement to reload distance matrix                      @
hexdist = sqrt( (hex_for[.,4]-hex_for[.,4]')^2 +
                (hex_for[.,5]-hex_for[.,5]')^2 );
save path=^pathname hexdist;
load path=^pathname hexdist;

@ Load matrix of compartment treatment schedules @
load path=^pathname bestcode;


@ Load these procs here because they use global variables specified above @
#include "d:\\pnwr\\procs2.g";


@ Distances matrix:  HEX_ID, Distances to stream and property boundaries @
dist = hex_attr[.,1 5:9];

@ If pine components are missing in a hexagon (area=0) convert age, BA,
    and canopy closure values to zero @
hex_for[.,17:19 21:23] = missrv(hex_for[.,17:19 21:23],0);

@ Make a list of candidate cells for recruitment clusters.  Remove all  @
@   cells occurring on refuge edge or too close to streams.             @
cand0 = hex_for[.,1] ~ ones(hexes,1);

@   Eliminate portions of hexagons found on refuge edge @
```

```
cand0[.,2] = cand0[.,2].*(hex_for[.,6]/fullsize.>=0.9999);

@   Eliminate hexagons too close to streams @
cand0[.,2] = cand0[.,2].*(dist[.,2].>=strmdist);

@   Append indicators in CAND0 to HEX_FOR;  @
hex_for = hex_for ~ cand0[.,2];

@ Times of last burning and thinning will be tracked separately in
    each pine stand.  Move TRTYEAR variable (col 8) to end of HEX_FOR
    (new col 27), put copies of LASTBURN (col 10) into cols 8-9, and
    put copies of LASTTHIN (col 11) into cols 10-11.                   @
hex_for = hex_for ~ hex_for[.,8];
hex_for[.,8] = hex_for[.,10];
hex_for[.,9] = hex_for[.,10];
hex_for[.,10] = hex_for[.,11];

@ Replace year of last thinning with age at last thinning.  For stands that
    were P1 at time of last treatment, age at last thinning = 0.           @
lastthin = zeros(hexes,2);
ageattrt = hex_for[.,17 21] - (start_yr - hex_for[.,10 11]);
not_p1 = (ageattrt.>=16);
lastthin = substute(lastthin,not_p1,ageattrt);
hex_for[.,10 11] = lastthin;

@ Replace year of last burning with age at last burning.  Maximum time
    since burn = 10 years.                                                 @
  @ If year of last burn = 0, then last burn occurred more than 10 years ago.
      Replace non-zero year values with age at last burn.                  @
ageattrt = substute(hex_for[.,8 9],(hex_for[.,8 9].>0),
                    hex_for[.,17 21] - (start_yr - hex_for[.,8 9]));
ageattrt = substute(ageattrt,(ageattrt.<16),0);
hex_for[.,8 9] = ageattrt;



@ Hardwood encroachment into initial time period @
hex_for = hwdcreep(hex_for);

@ Project forest state into the initial time period @
hex_for = for_grow(hex_for,hex_attr);

@ Forest disturbance at initial time period @
```

Appendix C.12. Spatially-explicit Refuge simulation model. Continued.

```
hex_for = disturb(hex_for);

@ Project understory state given projected overstory state @
veg = undstory(hex_for,hex_attr);

@print "Initial RCW state " (sumc(hex_for[.,24]));@

@ Project RCW state into initial time period @
hex_for = rcw_grow(hex_for,hex_attr,veg,prodmod);

@ Project WOTH state into initial time period @
woth = wothdens(hex_for,veg);



proc (1) = time_sim(decis,burn);
  local forest,t,len,vegdens,top_n,rcwactiv,rcwquota,comps,timevec,
    thistime,looptime,pine1,pine2,uh,p1,p2,p3,p4,p1_tot,p2_tot,p3_tot,p4_tot,
    uh_tot,for_tot,statevec,result,compsel;

  format 10,6;

  @ Initialize FOREST to HEX_FOR at beginning of each time projection @
  forest = hex_for;

  len = rows(decis);
  compsel = (forest[.,7].==seqa(1,1,34)');

  t = 0;
  vegdens = undstory(forest,hex_attr);
  woth = wothdens(forest,vegdens);
  result = t~sumc(forest[.,24])~sumc(woth);
  timevec = zeros(10,1);
  thistime = timevec;
  looptime = date;
  do while (t<len);
    t = t+1;

    @ Select groups of compartments for this year @
    comps = kgrpselr(allcomp,decis[t,2],decis[t,1])';

    @ Select patches in COMPS for regeneration @
    forest = regencut(forest,comps);
```

```
thistime[1] = ethsec(looptime,date);
looptime = date;

@ Compute vegetation state prior to thinning @
vegdens = undstory(forest,hex_attr);
thistime[2] = ethsec(looptime,date);
looptime = date;

@ Carry out thinning in compartments @
forest = thinning(forest,hex_attr,comps);
thistime[3] = ethsec(looptime,date);
looptime = date;

@ Carry out burning in compartments selected from refuge at large @
{forest,top_n} = burning(forest,vegdens,burn);
thistime[4] = ethsec(looptime,date);
looptime = date;

rcwactiv = sumc(forest[.,24]);
@ Update the RCW recruitment quota every 5 years @
if t==1 or t%5==0;
  rcwquota = clusgrow*rcwactiv;
endif;

@ Place RCW recruitment clusters @
forest = recrclus(forest,hex_attr,rcwquota);
thistime[5] = ethsec(looptime,date);
looptime = date;

@ Hardwood encroachment @
forest = hwdcreep(forest);
thistime[6] = ethsec(looptime,date);
looptime = date;

@ Project forest state into the next time period @
forest = for_grow(forest,hex_attr);
thistime[7] = ethsec(looptime,date);
looptime = date;

@ Forest disturbance @
forest = disturb(forest);
thistime[8] = ethsec(looptime,date);
looptime = date;
```

```
    @ Project understory state given future overstory state @
    vegdens = undstory(forest,hex_attr);
    thistime[9] = ethsec(looptime,date);
    looptime = date;

    @ Project RCW state into next time period @
    forest = rcw_grow(forest,hex_attr,vegdens,prodmod);
    thistime[10] = ethsec(looptime,date);
    looptime = date;

    @ Project WOTH state into next time period @
    woth = wothdens(forest,vegdens);

    result = result | (t~sumc(sumc(compsel.*forest[.,24])~woth)');

    timevec = timevec + thistime;

  endo;

  timevec = timevec/len/100;

  retp(result);

endp;


now1 = date;
now2 = now1;
outcomes = {};
i = 0;
do while i<reps;
  i = i+1;
  j = 0;
  do while j<strings;
    j = j+1;

    @ Construct decision vector @
    decis = bestcode[j,.];
    n_reps = ones(1,ceil(timemax/cols(decis))+1);
    decis = n_reps*~decis;
    decis = decis[startgrp:startgrp+timemax-1]';
    groups = n_reps*~maxcomp';
    groups = groups[startgrp:startgrp+timemax-1]';
    times = seqa(start_yr+1,1,timemax);
    decis = decis~groups;

    k = 0;
    do while k<rows(burn_max);
      k = k+1;
      burn = burn_max[k];
      format /rdn 4,0;
      print "Rep " i ", Decis " j ", Burn " burn;;
      result = time_sim(decis,burn);
      format /rdn 6,0;
      print ", Active RCW & WOTH = " (sumc(result)');;
      timer1 = ethsec(now1,date)/100/60;
      timer2 = ethsec(now2,date)/100/60;
      now2 = date;
      format /rdn 6,1;
      print ", Sim time " timer2 " Elap time " timer1;
      outcomes = outcomes | (i~j~burn~vec(result)');
      save path=^pathname outcomes;
    endo;
  endo;
endo;

etstr(ethsec(begtime,date));
```

PROCS1.G

```
@  PROCS1.G  @

proc i_to_p(ivector,steps,base_p,power);
  @
    Given a K-vector (IVECTOR) of index values drawn from a rectangular
      grid of dimension K, each dimension indexed by i=(1, 2, ..., STEPS),
      I_TO_P converts the set of indices to a set of power-transformed
      proportions P.  The BASE_P parameter specifies the smallest value
      of the proportion when the index value is at its smallest value
      (i.e., i=1).  The POWER parameter specifies the degree of
      exponentiation to be applied to the component ratios in computing
      the logits.  This procedure is the inverse of P_TO_I.
```

```
  @
  local numcomps,half,blogit,scale,diff,signs,denom,p,logits;

  numcomps = rows(ivector)+1;
  half = (steps+1)/2;
  blogit = ln(base_p/(1-(numcomps-1)*base_p));
  scale = -((-(1.-half))^power)/blogit;

  @ Read current state variables and convert them to logits @
  diff = ivector-half;
  signs = -(diff.<0) + (diff.>=0);
  logits = signs.*(signs.*diff)^power/scale;
  denom = 1+sumc(exp(logits));

  @ Calculate proportions @
  p = ( exp(logits) | 1 ) / denom;
  retp(p);

endp;


proc p_to_i(pvector,steps,base_p,power);
  @
    Inverse of procedure I_TO_P.  See I_TO_P for explanation of
      parameters.
  @
  local numcomps,half,blogit,scale,logits,signs,i;

  numcomps = rows(pvector);
  half = (steps+1)/2;
  blogit = ln(base_p/(1-(numcomps-1)*base_p));
  scale = -((-(1.-half))^power)/blogit;

  logits = ln(pvector[1:(numcomps-1)]/pvector[numcomps]);
  signs = -(logits.<0) + (logits.>=0);

  @ Calculate indices @
  i = (abs(logits)*scale)^(1/power).*signs + half;
  retp(i);

endp;
```

```
proc dlookup(x,dmatrix,transfrm);
  @
    Looks up decision value vector from decision table given current
        system state.  System state is first transformed to index state, then
        look-up is performed on index state.
  @
  local base_p,power,steps,c,y,dec,x_i,q,dec1;

  steps = transfrm[1];
  base_p = transfrm[2];
  power = transfrm[3];

  @ Calculate index state from system state @
  x_i = p_to_i(x,steps,base_p,power);
  c = rows(x_i);

  @ Round up values less than 1 @
  x_i = x_i + (x_i.<=1);

  @ Round down values greater than STEPS @
  x_i = x_i + steps*(x_i.>=steps);

  @ Round off other values to nearest integer @
  x_i = (round(x_i))';

  q = seqa(c-1,-1,c);
  q = steps^q;
  q = (x_i-1)*q+1;

  dec = dmatrix[q,(c+1):cols(dmatrix)];

  retp(dec');

endp;


proc (1) = combin(n,k);
  /****************************************************************************
  Computes combinatorial function for integers N and K, returns integer COMBIN.
  ****************************************************************************/
  local combin,diff,x,i;

  if k==0 or k==n;
```

Appendix C.12. Spatially-explicit Refuge simulation model. Continued.

```
    combin = 1;
  else;
    diff = n-k;
    if k<diff;
      x = k;
    else;
      x = diff;
    endif;
    combin = 1;
    i = 0;
    do while i<x;
      i = i+1;
      combin = combin*((n-i+1)/i);
    endo;
  endif;
  combin = round(combin);
  retp(combin);

endp;


proc (1) = kgrpselr(n,k,r);
  /**************************************************************************
  Provides the R-th ordered combination of N objects chosen K at a time.
    Valid values of R are 0, 1,..., COMBIN(N,K)-1.
    (ref: Algorithm 2.8 (p. 45) of Kreher & Stinson 1999)
  **************************************************************************/
  local krow,x,i,z;

  if k==0;
    krow = {};
  else;
    krow = zeros(1,k);
    x = 1;
    i = 0;
    do while i<k;
      i = i+1;
      z = combin(n-x,k-i);
      do while z<=r;
        r = r-z;
        x = x+1;
        z = combin(n-x,k-i);
      endo;
```

```
      krow[1,i] = x;
      x = x+1;
    endo;
  endif;
  retp(krow);

endp;



proc (1) = rnk_kgrp(n,x,sorted);
  /**************************************************************************
  Compute the rank order R of a vector of objects chosen K at a time from N
    total objects.  X is a row vector of size K.  If X is already sorted,
    set SORTED = 1; otherwise, SORTED = 0.  RNK_KGRP is the inverse
    procedure of KGRPSELR, i.e., R = RNK_KGRP(N,KGRPSELR(N,K,R),SORTED).
  **************************************************************************/
  local y,k,r,i,npart,kpart,j,m;

  y = x';
  if not sorted;
    y = sortc(x',1);
  endif;
  k = rows(y);
  r = 0;
  i = 0;
  do until i==k;
    i = i+1;
    npart = n-i;
    kpart = k-i;
    j = y[i];
    m = i;
    do until m==j;
      npart = npart-1;
      m = m+1;
      r = r + combin(npart,kpart);
    endo;
  endo;

  retp(r);

endp;
```

```
proc (1) = ksuccess(t,n,k);
  /**************************************************************************
  For vector T representing a single draw of K of N objects, N>K, this
    procedure provides vector U, representing the next rank-ordered draw--
    i.e., T = KGRPSELR(N,K,R) and U = KGRPSELR(N,K,R+1)
    (ref. Algorithm 2.6 (p. 43) of Kreher & Stinson 1999)
  **************************************************************************/
  local u,i,j;

  u = t;
  i = k;
  j = (i==0) + i*(i>0);       @ So that T vector is not given a zero index @
  do while i>=1 and t[j]==n-k+i;
    i = i-1;
    j = (i==0) + i*(i>0);
  endo;
  if i==0;
    u = {};
  else;
    j = i-1;
    do while j<k;
      j = j+1;
      u[j] = t[i]+1+j-i;
    endo;
  endif;
  retp(u);
endp;


proc (1) = uniqmat(x,columns,solitary,tol);

  /***  Procedure UNIQMAT  ****************************************************
    Inputs:  X        = N x K matrix
             COLUMNS  = P x 1 vector
             SOLITARY = 0/1 scalar
             TOL      = scalar
    Output:  M x (K+1) matrix, M<=N

    Removes duplicates of records in X.  Duplicated records are those which
    have the same values (within tolerance level TOL) in columns indicated by
    COLUMNS variable.  If SOLITARY=0, one record of a set of duplicates is
    retained.  If SOLITARY=1, all records in a duplicated set are removed
    (i.e., only unique records in X are retained).  Matrix returned has M rows,
    M <= N.  Return matrix is sorted by the target columns, but an extra column
    (column P+1) stores the original row index of X.  From this column, one can
    determine the sort order of X and which rows were removed from X.
  ****************************************************************************/

  local y,ycount,i,count,target,uniq,j,xcol,xindex,colindx;

  xindex = seqa(1,1,rows(x));
  xcol = cols(x);

  y = x[.,columns];            /** Round sorting columns to tolerance level **/
  y = round(y/tol)*tol;
  x = x~xindex~y;      /** Append row index column XINDEX and rounded values **/

  colindx = seqa(1,1,cols(x));
  colindx = selif(colindx,colindx.>xcol+1);
  x = sortmc(x,colindx);                     /**  Sort X by rounded values  **/

  y = {};
  ycount = {};
  i = 0;
  count = 0;
  do while i<rows(x);   /**  Travel down rows of X looking for dups **/
    i = i+1;
    count = count+1;
    target = x[i,columns];    /**  Pull out target row,  **/
    y = y|x[i,.];             /**  augment output matrix Y **/
    uniq = 0;                 /**  Assume target has duplicates below **/
    j = i;
    do until uniq==1;              /** Scan rows below for 1st nonduplicate **/
      if j==rows(x);                    /** Condition for end of file **/
        ycount = ycount|count;
        i = rows(x);
        uniq = 1;
      else;                             /** Not end of file **/
        j = j+1;
        if sumc((abs(target-x[j,columns]).>tol)');
                                           /** Found first nonduplicate **/
          uniq = 1;
          i = j-1;
          ycount = ycount|count;
        else;                              /** Found duplicate **/
```

Appendix C.12. Spatially-explicit Refuge simulation model. Continued.

```
          count = count+1;                    /** Increment counter on target*/
        endif;
      endif;
    endo;                          /** Found nonduplicate, or end of file **/
    if uniq==1;
      count = 0;
    endif;
  endo;                 /**  Next target row in X **/
  if solitary==1;              /** If SOLITARY=1, **/
    y = selif(y,ycount.==1);     /** remove all in duplicated set **/
  endif;
  retp(y);              /**  Return output matrix **/
endp;


proc (1) = gridindx(xy,minmax);

  /***  Procedure GRIDINDX  ***************************************************
    Inputs:  XY     = N x 2 matrix
             MINMAX = 2 x 2 matrix
    Output:  N x 1 vector

    Converts I,J addresses for N cells on a 2-dimensional grid into a
    sequential (1-dimensional) address.  Addresses are contained in matrix XY.
    MINMAX contains grid dimensions:  [1,1]=column (I) min, [1,2]=row (J) min,
    [2,1] = column (I) max, [2,2] = row (J) max.  Inverse of INDXGRID
    procedure.
  ****************************************************************************/

  local g;

  g = round((minmax[2,1]-minmax[1,1]+1)*(xy[.,2]-1) + xy[.,1]);

  retp(g);

endp;


proc (1) = indxgrid(i,minmax);

  /***  Procedure INDXGRID  ***************************************************
    Inputs:  I      = N x 1 vector
             MINMAX = 2 x 2 matrix
```

```
    Output:  N x 2 vector

    Converts N 1-dimensional sequential addresses to I,J addresses on a
    2-dimensional spatial grid.  MINMAX contains grid dimensions:
    [1,1]=column (I) min, [1,2]=row (J) min, [2,1] = column (I) max,
    [2,2] = row (J) max.  Inverse of GRIDINDX procedure.
  ****************************************************************************/

  local x,y;

  x = i%(minmax[2,1]-minmax[1,1]+1);
  x = recode(x,(x.==0),minmax[2,1]);
  y = round((i-x)/(minmax[2,1]-minmax[1,1]+1) + 1);

  retp(x~y);

endp;


@  Function GRIDXY -- Creates a matrix of (X,Y) coordinate pairs for a grid  @
@   of row size R (Y values) and column size C (X values).                   @
fn gridxy(r,c) = seqa(1,1,c).*.ones(r,1) ~ ones(c,1).*.seqa(1,1,r);


proc (1) = ring(x,y,minmax,n);

  /***  Procedure RING *********************************************************
    Inputs:  X,Y    = scalar
             MINMAX = 2 x 2 matrix
             N      = scalar, value = {6, 12, 48}
    Output:  K x 2 matrix, K<=SIZE

    Given a cell address (X,Y), RING returns I,J addresses of the N nearest-
    neighboring cells of (X,Y).  MINMAX contains values that defines the extent
    of the spatial grid:  [1,1]=column (I) min, [1,2]=row (J) min,
    [2,1] = column (I) max, [2,2] = row (J) max.  RING returns N addresses
    if the neighborhood ring does not cross a grid boundary, otherwise, RING
    returns some number smaller than N.
  ****************************************************************************/

  local x_vec,x_odd,x_even,r,e1,e2,e3,e4,z;

  @ X_VEC vectors are offsets in the X direction, X_ODD are offsets   @
```

Appendix C.12. Spatially-explicit Refuge simulation model. Continued.

```
@  in the Y direction when X is odd, and X_EVEN are offsets in the  @
@  Y direction when X is even.                                      @

if (n==6);
  x_vec  = {-1  0  1  1  0 -1};
  x_odd  = { 0  1  0 -1 -1 -1};
  x_even = { 0 -1  0  1  1  1};

elseif (n==12);
  x_vec  = {-1 -1  0  1  1  2  1  1  0 -1 -1 -2};
  x_odd  = { 0  1  1  1  0  0 -1 -2 -1 -2 -1  0};
  x_even = { 1  2  1  2  1  0  0 -1 -1 -1  0  0};

elseif (n==48);
  x_vec  = {-1 -1  0  1  1  2  1  1  0 -1 -1 -2
            -2 -3 -3 -2 -1 -1  0  0  1  1  2  3  2  3  4  3  3  4
             2  3  3  2  1  1  0  0 -1 -1 -2 -3 -2 -3 -4 -3 -3 -4};
  x_odd  = { 0  1  1  1  0  0 -1 -2 -1 -2 -1  0
             1  1  2  2  2  3  2  3  3  2  2  2  1  1  1  0 -1 -1
            -1 -2 -3 -2 -3 -4 -2 -3 -4 -3 -2 -3 -1 -2 -1 -1  0  1};
  x_even = { 1  2  1  2  1  0  0 -1 -1 -1  0  0
             1  2  3  2  3  4  2  3  4  3  2  3  1  2  1  1  0 -1
            -1 -1 -2 -2 -2 -3 -2 -3 -3 -2 -2 -2 -1 -1 -1  0  1  1};

endif;

x_odd = (x_vec | x_odd)';
x_even = (x_vec | x_even)';

@ Calculate coordinates of cells in ring @
if (x%2<0.001);      @ X even @
  r = (x~y) + x_even;
else;                @ X odd @
  r = (x~y) + x_odd;
endif;

@ Test for cells outside of grid boundaries @
e1 = (minmax[1,1].>r[.,1]);
e2 = (minmax[1,2].>r[.,2]);
e3 = (minmax[2,1].<r[.,1]);
e4 = (minmax[2,2].<r[.,2]);

@ Replace coordinate values for out-of-bounds cells with grid limit values @
```

```
  r = substute(r,e1~e2,minmax[1,.]);
  r = substute(r,e3~e4,minmax[2,.]);

  z = gridindx(r,minmax);

  @ Remove duplicate (out-of-bound) cells @
  r = r[uniqindx(z,1),.];

  @ Remove focal cell from set of addresses @
  r = delif(r,r[.,1].==x .and r[.,2].==y);

  retp(r);

endp;


/*
** BETADIST.SRC
**   Beta distribution procedures:
**     BETAPARM -- converts between (mean, variance) and (alpha, beta)
**                     parameterizations
**     BETA_INV -- calculates inverse of beta CDF
**     BETA_EXP -- calculates expected value of beta PDF over selected domain
**
**  Clint Moore
**  8 March 2001
*/

/*
**  Proc BETAPARM
**
**  Converts between alpha/beta and mean/variance parameterizations of the
**    beta distribution.  Alpha and beta inputs must both be >0.  Mean (mu)
**    inputs must be 0<mu<1, and variance (var) inputs must be >0.  Also,
**    (1-mu)*mu > var.
**
**  Inputs:  matrix PARMS (Nx2), column 1 -- alpha, or mean parameter
**                               column 2 -- beta, or variance parameter
**           scalar SWITCH,      SWITCH=0 -- convert alpha/beta to mean/var
**                               SWITCH=1 -- convert mean/var to alpha/beta
**
**  Output:  matrix (Nx4), converted parameter values
**                            columns 1-4 -- alpha, beta, mean, variance
```

```
**
**  Usage:   X = BETAPARM(VALS,0);
*/
proc betaparm(parms,switch);
  local y,p1,p2,sum;

  p1 = parms[.,1];
  p2 = parms[.,2];
  y = parms;

  if switch==0;        @ alpha/beta to mean/var @
    sum = p1+p2;
    y[.,1] = p1 ./ sum;
    y[.,2] = (p1.*p2) ./ ( sum.*sum.*(sum+1) );
    y = parms~y;

  elseif switch==1;    @ mean/var to alpha/beta @
    y[.,1] = ( (1-p1).*(p1./p2) - 1 ) .* p1;
    y[.,2] = ( p1.*p1.*(p1-2) + p1.*(1+p2) - p2 ) ./ p2;
    y = y~parms;

  else;                @ null matrix returned if SWITCH not correct @
    y = {};
  endif;

  retp(y);
endp;


/*
**  Proc BETA_INV
**
**  Calculates inverse of beta CDF by simple grid search.  Adjust TOL and
**     IMAX values to change solution precision.
**
**  Inputs:  matrix P (RxC),    probability values
**           scalar ALPHA
**           scalar BETA
**                            -1
**  Output:  matrix (RxC), CDF  (P,APHA,BETA)
**
**  Usage:   X = BETA_INV(P,ALPHA,BETA);
*/
```

```
proc (1) = beta_inv(p,alpha,beta);
  local flag,i,imax,step,val,tol,test,diff,pvec,j,valmat,q;

    @ Execution control parameters @
  imax = 150;    @ maximum number of search iterations @
  tol = 1e-12;   @ tolerance value @
    @ @

  pvec = vecr(p);
  valmat = -1*ones(rows(pvec),1);
  j = 0;
  do while j<rows(pvec);

    j = j+1;
    q = pvec[j];

    if q==0 or q==1;   @ skip iterations if solution is known @
      val = q;

    else;
      step = 0.5;    @ initial step size @
      val = 0.5;     @ initial guess @
      flag = 0;
      i = 0;
      test = cdfbeta(val,alpha,beta);
      diff = q-test;

      do while (abs(diff)>tol and i<imax);
        i = i+1;
        if i==imax;         @ iteration limit reached @
          flag = 1;
        endif;
        step = step/2;      @ cut step size in half @
        if diff>0;          @ guess was too low, increase value @
          val = val+step;
        else;               @ guess was too high, decrease value @
          val = val-step;
        endif;
        test = cdfbeta(val,alpha,beta);
        diff = q-test;
      endo;

      if flag==1;           @ procedure didn't converge @
```

```
        if val>0.9999;           @ set value=1 if close to 1 @
          val = 1;
        elseif val<0.0001;       @ set value=0 if close to 0 @
          val = 0;
        endif;                   @ otherwise leave value as is @
      endif;

    endif;

    valmat[j] = val;

  endo;

  valmat = reshape(valmat,rows(p),cols(p));

  retp(valmat);
endp;


/*
**  Proc BETA_EXP
**
**  Calculates expected value of beta random variable within a user-specified
**     domain (0<= domain limits <=1).
**
**  Inputs:  scalar ALPHA,
**           scalar BETA,
**           matrix LIMS (2xN),  upper (top row) and lower (bottom row) limits
**                                   of integration, N limit pairs
**
**  Output:  vector (N), expected value of beta r.v. for each limit pair
**
**  Usage:   X = BETA_EXP(ALPHA,BETA,LIMS);
*/

  @ temporary values of __PARM1 and __PARM2 @
__parm1 = 1;
__parm2 = 1;

  @ expectation function @
proc __xbeta(x);
  retp( x .* x^(__parm1-1) .* (1-x)^(__parm2-1) );
endp;
```

```
proc beta_exp(alpha,beta,lims);
  local coeff,gam_a,gam_b,gam_ab,scale,y;

  __parm1 = alpha;
  __parm2 = beta;
  _intord = 40;

    @ Use gamma function for small values and lnfact for large values @
  if alpha<=1;
    gam_a = ln(gamma(alpha));
  else;
    gam_a = lnfact(alpha-1);
  endif;
  if beta<=1;
    gam_b = ln(gamma(beta));
  else;
    gam_b = lnfact(beta-1);
  endif;
  if alpha+beta<=1;
    gam_ab = ln(gamma(alpha+beta));
  else;
    gam_ab = lnfact(alpha+beta-1);
  endif;
  coeff = exp(gam_ab-gam_a-gam_b);

    @ Calculate scaling factors @
  scale = ( cdfbeta(lims[1,.],alpha,beta) - cdfbeta(lims[2,.],alpha,beta) )';

  y = intquad1(&__xbeta,lims) * coeff ./ scale;

  retp(y);
endp;
```

PROCS2.G

```
proc (1) = wothdens(f_state,veg);
  /***************************************************************************
    Estimates wood thrush population given forest overstory and understory
      attributes.  Procedure takes as input F_STATE, the current state of the
```

```
    forest hexagon grid, and VEG, the vector of hexagon understory
    vegetation densities.  Program returns WOTH, a vector of compartment-
    specific estimated values of wood thrush population size.
Globals:
  WOTH_P1, WOTH_P2, WOTH_P3, WOTH_P4
  WOTH_P3V, WOTH_P4V
***************************************************************************/
local pineonly,avg_ba,avg_cc,avg_tb,avg_veg,comps,sumarea,hardarea,
      pinearea,lowveg,veglogit,woth_h,woth_p,woth,weights,x,v,est;

@ Total hardwood areas within compartments @
comps = dummybr(f_state[.,7],seqa(1,1,34));
hardarea = comps'*(f_state[.,12]+f_state[.,13]);  @ Total hardwood area @

@ Calculate avg BA and canopy closure in all non-P1 stands by compartment @
@   Cols in PINEONLY are HEX_ID, COMP, ages of last burn, stands info, VEG @
pineonly = selif(f_state[.,1 7 8:9 16:23]~veg,
                       (f_state[.,16].>0 .and f_state[.,17].>=16) .or
                       (f_state[.,20].>0 .and f_state[.,21].>=16) );

@ Don't count P1 stands in area calculation @
pineonly[.,5 9] = substute(pineonly[.,5 9],pineonly[.,6 10].<16,0);

@ Total pine area in cells @
sumarea = pineonly[.,5]+pineonly[.,9];

@   Area-weighted average of pine basal area @
avg_ba = (pineonly[.,5].*pineonly[.,7] + pineonly[.,9].*pineonly[.,11]) ./
            sumarea;

@   Area-weighted average of canopy closure @
avg_cc = (pineonly[.,5].*pineonly[.,8] + pineonly[.,9].*pineonly[.,12]) ./
            sumarea;
@   Area-weighted average of time since last burn @
avg_tb = pineonly[.,6 10]-pineonly[.,3 4];
avg_tb = substute(avg_tb,avg_tb.>10,10);  @ Max time since burn = 10 yrs @
avg_tb = (pineonly[.,5].*avg_tb[.,1] + pineonly[.,9].*avg_tb[.,2]) ./
            sumarea;

@ Design matrix for compartment averages @
comps = dummybr(pineonly[.,2],seqa(1,1,34));
weights = comps.*sumarea;
pinearea = sumc(weights);               @ Total pine area in compartments @
```

```
sumarea = substute(pinearea,pinearea.==0,1);    @ Avoid zero denominators @

@ Compartment weighted means of BA, converted to ft^2/ac @
avg_ba = (weights'*avg_ba)./sumarea*4.356;

@ Compartment weighted means of canopy closure, convert to proportion @
avg_cc = (weights'*avg_cc)./sumarea/100;

@ Compartment weighted means of time since burning @
avg_tb = (weights'*avg_tb)./sumarea;

@ Compartment weighted means of vegetation density @
avg_veg = (weights'*pineonly[.,13])./sumarea;

@ Estimate compartment averages of low veg density @
lowveg = woth_p1[1] + woth_p1[2]*avg_tb;
@   Estimates must fall between 0, 1 @
lowveg = substute(lowveg,lowveg.>1,1);
lowveg = substute(lowveg,lowveg.<0,0);

@ Estimate compartment averages of veg density logits @
veglogit = woth_p2[1] + woth_p2[2]*avg_tb + woth_p2[3]*avg_veg;

if wothmod==2;
  @ Apply limits to model inputs @
  avg_ba = substute(avg_ba,avg_ba.<ba_lims[1],ba_lims[1]);
  avg_ba = substute(avg_ba,avg_ba.>ba_lims[2],ba_lims[2]);
  avg_cc = substute(avg_cc,avg_cc.<cc_lims[1],cc_lims[1]);
  avg_cc = substute(avg_cc,avg_cc.>cc_lims[2],cc_lims[2]);
  lowveg = substute(lowveg,lowveg.<low_lims[1],low_lims[1]);
  lowveg = substute(lowveg,lowveg.>low_lims[2],low_lims[2]);
  veglogit = substute(veglogit,veglogit.<log_lims[1],log_lims[1]);
  veglogit = substute(veglogit,veglogit.>log_lims[2],log_lims[2]);
endif;

@ Estimate density of WOTH in hardwood habitat @
x = ones(rows(pinearea),1) ~ avg_ba ~ avg_cc ~ lowveg ~ avg_ba.*avg_cc;
est = x*woth_p3';
v = diag(x*woth_p3v*x');
est = est+rndn(rows(est),1).*v;
woth_h = exp(est+v/2);

@ Estimate density of WOTH in pine habitat @
```

Appendix C.12. Spatially-explicit Refuge simulation model. Continued.

```
  x = x[.,1]~avg_ba ~ lowveg ~ veglogit ~ avg_ba.*veglogit ~ lowveg.*veglogit;
  est = x*woth_p4';
  v = diag(x*woth_p4v*x');
  est = est+rndn(rows(est),1).*v;
  woth_p = exp(est+v/2);

  @ Calculate total WOTH population @
  woth = (woth_h.*hardarea + woth_p.*pinearea)/10000;

  retp(woth);

endp;


proc (1) = vegtoba(veg,parms);
  /***************************************************************************
    Given a vector of vegetation density values and a vector of tranformation
      parameters, procedure converts veg values into estimated values of
      total midstory BA.  Procedure uses values reported by Loeb et al. to
      normalize a transformation (x^r) of veg density.  Parameters in PARMS are
      (1) an exponent r, (2) a veg density mean, and (3) a BA mean.  The SD
      for veg density is enforced by assuming equal CV between veg density and
      BA.  Midstory BA is returned in units of m^2/ha.
  ***************************************************************************/
  local r,ba;


  r = parms[1];
  veg = veg^r;
  ba = (veg-parms[2])/(parms[2]/parms[3]) + parms[3];
  ba = ba/4.356;

  retp(ba);

endp;


proc (1) = rcw_grow(f_state,f_attr,veg,prodmod);
  /***************************************************************************
    Projects RCW cluster activity from one time period to the next, given
      forest state information.  Program accepts as input F_STATE, the
      current state of the forest hexagon grid, F_ATTR, the matrix of
      permanent forest attributes, VEG, the vector of hexagon understory
    vegetation densities, and PRODMOD, an integer indicator of production
    model alternative 1 (PRODMOD=1) or 2 (PRODMOD=2).  Program returns F_OUT,
    a copy of F_STATE with the RCW cluster activity column (24) indicated
    as either active or inactive.
  Globals:
    OUT800
    MIDBA
    LOEBPARM
    MODEL1, MODEL2
    GRIDSIZE
    FH1_AGE, FH1_BA1, FH1_BA2
    RECRPARM
***************************************************************************/
local f_out,owner,amt800,pine1,pine2,fullsize,rcw,area1,age1,area2,age2,
      gone,mba,logit,prob,new_occ,row,col,clutch,hab,cats,probs,cand800,
      pullamt,amt,i,habsum,ring_xy,pullmat,sel1,sel2,ringarea,
      habitat,tothab,habcat,prodprob,randraw,prodidx,prod,inactiv,activ,
      dist,isol,new_recr,rans,h,w,v,tol,idx,
      theta,tau,psi,f_i,sum_f_i,p,q,p2,p3,p4,q2,q3,q4,hef,mba2;

f_out = f_state;

@ RCW matrix:  HEX_ID, row/col, X/Y, cluster occurrence, cluster activity @
rcw = f_state[.,1:5] ~ (f_state[.,25].>0) ~ f_state[.,24];

@ If no active clusters, leave procedure @
if sumc(rcw[.,7]).==0;
  goto nomore;
endif;

@ Ownership matrix:  HEX_ID, 4 ownership proportions in each of 4 buffers @
owner = f_attr[.,1 10:25];

@ Estimated proportion (based on ownership and estimated provision)
    of foraging habitat on non-refuge land in circular buffers @
amt800 = owner[.,7:9]*out800;

@ PINE matrices for stands 1 and 2:  HEX_ID, AREA, AGE, BA, and CC @
pine1 = f_state[.,1 16:19];
pine2 = f_state[.,1 20:23];

@ Full hexagon size @;
fullsize = maxc(f_state[.,6]);
```

Appendix C.12. Spatially-explicit Refuge simulation model. Continued.

```
@ Cell dimensions @
h = sqrt(fullsize/(2*sqrt(3)));
w = h*sqrt(3)/3;
v = 2*w;
tol = 0.01;

area1 = pine1[.,2];
age1 = pine1[.,3];
area2 = pine2[.,2];
age2 = pine2[.,3];

@ First check whether each present cluster site has been recently destroyed @
gone = (area1.>0).*(age1.==0).*(rcw[.,6].==1);
@   Change cluster status for such cells to "missing" and "inactive" @
rcw[.,6 7] = (1-gone).*rcw[.,6 7] + gone.*(0~0);

@ Convert veg density values at cluster site to midstory basal areas @
mba = vegtoba(veg,midba);

mba2 = ba_adj[1]*(mba-ba_adj[2]);

@ Apply Loeb et al. logit model to last year's active clusters to determine
    this year's activity @
  @ (1) Calculate probability of occupancy @
logit = loebparm[1] + loebparm[2]*mba2;
prob = exp(logit);
prob = prob./(1+prob);
  @ (2) This year's occupancy indicator is bernoulli with probability PROB @
new_occ = (rndu(rows(rcw),1).<prob);

rcw[.,7] = new_occ.*rcw[.,7];
  @ (3) Append logit value to RCW matrix (becomes 8th col of RCW) @
logit = loebparm[1] + loebparm[2]*mba;
rcw = rcw ~ logit;

theta = prodparm[1,prodmod];
tau = prodparm[2,prodmod];
clutch = nestling[.,1];

@ Calculate or estimate amount of foraging habitat in 800-m radius (HABSUM) @
@   CAND800 is subset of RCW (only hexes containing a cluster) @
cand800 = selif(rcw,rcw[.,6].==1);
```

```
if cand800/=miss(0,0);
  @ CAND800 is not empty @

  @   Pull corresponding values from AMT800 @
  pullamt = sumc(cand800[.,1].==(f_out[.,1]'));
  amt = selif(amt800,pullamt);

  @   Search over all the hexagons in CAND800 @
  i = 0;
  habsum = {};
  do while (i<rows(cand800));
    i = i+1;

    @ Find row index of this cell in F_STATE @
    idx = maxindc(cand800[i,1].==f_state[.,1]);

    @ Distance vector for this cluster @
    dist = hexdist[idx,.];

    @ Identify cells in 48-cell neighborhood, include focal cell @
    ring_xy = (dist.<(6*h)*(1+tol)) +
              (dist.>4*(v+w)*(1+tol)).*(dist.<(8*h)*(1-tol));
    ring_xy = ring_xy';

    @ Pull out both pine components for this ring @
    sel1 = selif(pine1,ring_xy);
    sel2 = selif(pine2,ring_xy);

    @ Calculate area of habitat ring @
    ringarea = sumc(selif(f_out[.,6],ring_xy));

    @ Calculate suitable habitat in ring @
    habitat = (sel1[.,3].>=fh1_age).*(sel1[.,4].>=fh1_ba1).*
                    (sel1[.,4].<=fh1_ba2).*sel1[.,2] +
              (sel2[.,3].>=fh1_age).*(sel2[.,4].>=fh1_ba1).*
                    (sel2[.,4].<=fh1_ba2).*sel2[.,2];
    tothab = sumc(habitat);

    @ Check for habitat extending beyond PNWR @
    if rows(sel1)<49 or ringarea/(49*fullsize)<0.9999;
      @ Estimate of total habitat found on and off-refuge @
      tothab = tothab + 49*fullsize*amt[i];
    endif;
```

Appendix C.12. Spatially-explicit Refuge simulation model. Continued.

```
  @ Convert foraging habitat area to hectares and append to HABSUM @
  habsum = habsum | tothab/10000;

endo;     @ Finished looping over all hexagons @

@ Project recruitment into inactive clusters based on productivity
    and distances between active and inactive clusters              @

if sumc(cand800[.,7])>0 and sumc(cand800[.,6])>sumc(cand800[.,7]);
  @ At least one active cluster and one inactive cluster on refuge @

  @ Productivity computation for active clusters @

  habsum = (habsum-fa_0)/fa_0;
  psi = theta*habsum + 1;
  f_i = nestling[.,2]'.*(psi^expon);
  sum_f_i = sumc(f_i');
  f_i = f_i./sum_f_i;
  q = 1./( 1+exp( logit_fl + tau*(psi-1) ) );
  p = 1-q;
  p2 = p.*p;
  p3 = p2.*p;
  p4 = p3.*p;
  q2 = q.*q;
  q3 = q2.*q;
  q4 = q3.*q;
  f_i[.,1] = f_i[.,1] + f_i[.,2].*q + f_i[.,3].*q2 + f_i[.,4].*q3
                                                   + f_i[.,5].*q4;
  f_i[.,2] = f_i[.,2].*p  + 2*f_i[.,3].*p.*q  + 3*f_i[.,4].*q2.*p
                                                   + 4*f_i[.,5].*q3.*p;
  f_i[.,3] = f_i[.,3].*p2 + 3*f_i[.,4].*p2.*q + 6*f_i[.,5].*q2.*p2;
  f_i[.,4] = f_i[.,4].*p3 + 4*f_i[.,5].*p3.*q;
  f_i[.,5] = f_i[.,5].*p4;
  prodprob = cumsumc(f_i')';

    @ (3) Draw random productivity values based on probabilities @
  randraw = rndu(rows(cand800),1);
  prodidx = (randraw.>prodprob)';
  prodidx = sumc(prodidx)+1;
  prod = clutch[prodidx];

    @ (4) Make sure that PROD=0 for inactive clusters @
  prod = 0*(cand800[.,7].==0) + prod.*(cand800[.,7].>0);
```

```
  @ (5) Append PROD to CAND800 (becomes 9th col of CAND800) @
cand800 = cand800 ~ prod;

@ Calculate recruitment probabilities for inactive clusters @

  @ (1) Segregate inactive from active clusters @
inactiv = selif(cand800,cand800[.,7].==0);
activ = selif(cand800,cand800[.,7].>0);

hef = activ[.,4 5 9] | (hef_x ~ hef_y ~ hef_prod);

  @ (2) Squared distances between active and inactive clusters (km^2) @
dist = ((inactiv[.,4]-hef[.,1]')^2 + (inactiv[.,5]-hef[.,2]')^2) /
       1000000;

  @ (3) Invert squared distances and multiply by productivity values @
isol = 1./dist;
isol = (hef[.,3]').*isol;

  @ (4) Sum over all active clusters for productivity isolation value @
isol = sumc(isol');
isol = 1./isol;

  @ (5) Combine with recruitment function parameters and add to logit
      (Add to logit only if ISOL exceeds threshold value)             @
logit = inactiv[.,8] +
             recrparm[2]*(isol-recrparm[1]).*(isol.>recrparm[1]);

  @ (6) Calculate probability of recruitment @
prob = exp(logit);
prob = prob./(1+prob);

  @ (7) Recruitment indicator is bernoulli with probability PROB @
rans = rndu(rows(prob),1);
new_recr = (rans.<prob);

if sumc(new_recr)>0;

  @ Identify inactive clusters in RCW that became active @
  inactiv = selif(inactiv[.,1],new_recr);
  pullmat = sumc(inactiv.==(rcw[.,1]'));

  @ Place vector of recruited clusters into RCW vector @
```

Appendix C.12. Spatially-explicit Refuge simulation model. Continued.

```
        rcw[.,7] = rcw[.,7] + pullmat;

      endif;

    endif;

  endif;

  @ Place revisions to cluster occurrence (RCW col 6) and cluster activity
      (RCW col 7) in appropriate columns in F_OUT forest status matrix @
  f_out[.,25] = rcw[.,6];  @ Cluster occurrence @
  f_out[.,24] = rcw[.,7];  @ Cluster activity @

  nomore:

  retp(f_out);

endp;



proc (1) = disturb(f_state);
  /***************************************************************************
    Pine areas and upland hardwood areas are altered by persistent rates of
      stochastic disturbance.  Parts of regenerated stands convert to
      hardwood.  Also, mature pine stands (P2-P4) are subject to random
      mortality events, and these are converted to regeneration.  Program
      accepts as input F_STATE, the matrix of forest cover attributes.
      Program creates F_OUT, which is copied from F_STATE but contains revised
      forest cover areas for UH and pine classes.
    Globals:
      M_RATES
      DSTRB_MN, DSTRB_B, DSTRB_HW
      BA0
  ***************************************************************************/
  local f_out,tot_area,rans,permute,index,disturb,sel,notsel,chg1,chg2,
        age1,age2;

  f_out = f_state;

  @ STAND MORTALITY @
  @ Draw random numbers for both pine stands @
  rans = rndu(rows(f_state),2);
```

```
@ Initialize mortality indicator vectors (0 = no mortality) @
chg1 = 0*rans[.,1];
chg2 = chg1;

@ Assign age-specific mortality rate thresholds to stand 1 @
age1 = (f_state[.,17].>=16 .and f_state[.,17].<40)~
       (f_state[.,17].>=40 .and f_state[.,17].<80)~
       (f_state[.,17].>=80);
chg1 = recode(chg1,age1,m_rates);
@ Randomly select stands to regenerate @
chg1 = (rans[.,1].<chg1);

@ Assign age-specific mortality rate thresholds to stand 2 @
age2 = (f_state[.,21].>=16 .and f_state[.,21].<40)~
       (f_state[.,21].>=40 .and f_state[.,21].<80)~
       (f_state[.,21].>=80);
chg2 = recode(chg2,age2,m_rates);
@ Randomly select stands to regenerate @
chg2 = (rans[.,2].<chg2);

@ Append vectors to form indicator matrices @
sel = chg1 ~ chg2;
notsel = 1-sel;

@ Change characteristics in forest state matrix for regenerated stands @
f_out[.,17 21] = notsel.*f_state[.,17 21] + sel*0;    @ Set ages to 0 @
f_out[.,18 22] = notsel.*f_state[.,18 22] + sel*ba0;  @ Set BAs to BA0 @
f_out[.,19 23] = notsel.*f_state[.,19 23] + sel*0;    @ Set CCs to 0 @
f_out[.,8 9]  = notsel.*f_state[.,8 9]  + sel*0;      @ Age/last burn = 0 @
f_out[.,10 11] = notsel.*f_state[.,10 11] + sel*0;    @ Age/last thin = 0 @

@ STAND DISTURBANCE @
@ Total area of refuge @
tot_area = sumc(f_state[.,6]);

@ Draw random disturbance area @
if dstrb_b==0;
  rans = dstrb_mn*tot_area;
else;
  rans = rndbeta(1,1,dstrb_b[1],dstrb_b[2])*tot_area;
endif;

@ Randomly permute rows of F_STATE @
```

```
permute = rndu(rows(f_state),1);
permute = permute~f_state[.,1 6];
permute = sortc(permute,1);

@ Cumulative sum of hexagon areas in permuted matrix @
permute = permute ~ cumsumc(permute[.,3]);

@ Difference between cumulative sum column and target total disturbance @
permute = permute ~ (permute[.,4]-rans);

@ Find row index of 1st positive difference (cumulative area > target) @
permute = permute ~ (permute[.,5].>=0);
permute = permute ~ cumsumc(permute[.,6]);
index = indexcat(permute[.,7],1);

@ Extract rows from 1 to INDEX @
disturb = permute[1:index,2];

@ Locate DISTURB hexagons in F_STATE @
sel = sumc(disturb.==f_state[.,1]');
notsel = 1-sel;

@ Remove DSTRB_HW proportion from upland hardwood @
f_out[.,13] = notsel.*f_state[.,13] +
                 sel.*(f_state[.,13] - dstrb_hw*f_state[.,13]);

@ Change pine stand characteristics @
@    Part of upland hardwood converts to pine regeneration in stand 1 @
f_out[.,16] = notsel.*f_state[.,16] +
                 sel.*(f_state[.,16] + dstrb_hw*f_state[.,13]);
@    Keep area for stand 2 as is @
f_out[.,20] = f_state[.,20];

@ Change characteristics in forest state matrix for regenerated stands @
@  (Use F_OUT because these columns were affected in mortality step)   @
f_out[.,17 21] = notsel.*f_out[.,17 21] + sel*0;    @ Set ages to 0 @
f_out[.,18 22] = notsel.*f_out[.,18 22] + sel*ba0;  @ Set BAs to BA0 @
f_out[.,19 23] = notsel.*f_out[.,19 23] + sel*0;    @ Set CCs to 0 @
f_out[.,8 9]  = notsel.*f_out[.,8 9]  + sel*0;      @ Age/last burn = 0 @
f_out[.,10 11] = notsel.*f_out[.,10 11] + sel*0;    @ Age/last thin = 0 @

retp(f_out);
```

```
endp;


proc (1) = hwdcreep(f_state);
  /***************************************************************************
    Pine areas and upland hardwood areas are altered by persistent (stochastic)
      rates of hardwood encroachment.  Program accepts as input F_STATE, the
      matrix of forest cover attributes.  Program creates F_OUT, which is
      copied from F_STATE but contains revised forest cover areas for UH and
      pine classes.
    Globals:
      BURNLIM, THINLIM
      HWD_P1, HWD_P2, HWD_P3, HWD_P4, HWD_BETA
  ***************************************************************************/
  local f_out,p1,p2,p3,p4,tsb,tst,burn,thin,convert,rans,i,k,c,age,area,
        a_burn,a_thin;

  f_out = f_state;

  @ Create fixed or random vectors of encroachment rates @
  @   8 columns--> 4 pine types for each of 2 stands in hexagon @
  if hwd_beta==0;
    rans = ones(rows(f_state),8) .* (hwd_p1~hwd_p2~hwd_p3~hwd_p4 ~
                                     hwd_p1~hwd_p2~hwd_p3~hwd_p4);
  else;
    rans = rndbeta(rows(f_state),1,hwd_beta[1],hwd_beta[2]) ~
           rndbeta(rows(f_state),1,hwd_beta[1],hwd_beta[2]);
    rans = rans[.,1] ~ (rans[.,1] .* ( (hwd_p2~hwd_p3~hwd_p4)/hwd_p1 ) ) ~
           rans[.,2] ~ (rans[.,2] .* ( (hwd_p2~hwd_p3~hwd_p4)/hwd_p1 ) );
  endif;

  @ Column indices for area, age, age/last burn, age/last thin for pine types @
  c = {16 17 8 10,
       20 21 9 11};

  @ Calculate amount of hardwood encroachment for both pine stands @
  i = 0;
  do while i<2;
    i = i+1;
    k = (i-1)*4;   @ Index for pine stand random variate group @

    @ Prepare age, area, age/last burn, age/last thin columns @
    area = f_state[.,c[i,1]];
```

Appendix C.12. Spatially-explicit Refuge simulation model. Continued.

```
    age = f_state[.,c[i,2]];
    a_burn = f_state[.,c[i,3]];
    a_thin = f_state[.,c[i,4]];

    @ Indicators of pine type in stand I @
    p1 = (area.>0 .and age.>0   .and age.<16);
    p2 = (area.>0 .and age.>=16 .and age.<40);
    p3 = (area.>0 .and age.>=40 .and age.<80);
    p4 = (area.>0 .and age.>=80);

    @ Calculate time since burning and time since thinning for non-P1 stands @
    tsb = age-a_burn;
    tsb = -1*(age.<16) + tsb.*(age.>=16);
    tst = age-a_thin;
    tst = -1*(age.<16) + tst.*(age.>=16);

    @ Indicates if stands have been burned and/or thinned in time window @
    burn = (area.>0 .and tsb.>=0 .and tsb.<=burnlim);
    thin = (area.>0 .and tst.>=0 .and tst.<=thinlim);

    @ Apply (random) rates of hardwood conversion @
    @   Prevent P2-P4 areas from converting if recently thinned or burned @
    convert = area.*( (1-burn).*(1-thin).*( (rans[.,2+k].*p2) +
                                            (rans[.,3+k].*p3) +
                                            (rans[.,4+k].*p4) ) +
                      rans[.,1+k].*p1 );

    @ Subtract converted amount from pine stand I, add to hardwood stand @
    f_out[.,c[i,1]] = area - convert;
    f_out[.,13] = f_out[.,13] + convert;

  endo;

  retp(f_out);

endp;



proc (2) = burning(f_state,veg,n);
  /***************************************************************************
    Selects compartments for burning given a maximum number of compartments
      that may be burned.  Program accepts for input F_STATE, the current
    state of the forest hexagon grid, VEG, the vector of hexagon understory
    vegetation densities, and N, the maximum number of compartments burned.
    Program returns F_OUT, a copy of F_STATE with age at last burning updated
    appropriately for burned stands, and COMPS, the compartments selected
    for burning.
  Globals:
    GRIDSIZE, FULLSIZE
***************************************************************************/
local f_out,comp_all,rcw,clus,non_clus,nest,forage,rcw_list,clusmat,
      compclus,vegdist,minveg,veges,pullmat,other,mostpine,combine,sorted,
      top_n,veg_comp,comps,h,w,v,tol,dist,ring,forveg,compdum,formeans;

f_out = f_state;
top_n = {};

@ If no burning is to be conducted, leave procedure @
if n==0;
  goto getout;
endif;

comp_all = seqa(1,1,34);

@ RCW matrix:  HEX_ID, X, Y, Occurrence of cluster, Activity of cluster @
rcw = f_state[.,1 4 5] ~ (f_state[.,25].>0) ~ f_state[.,24];

@ Find all cells containing an RCW cluster, attach vegetation measures @
clus = selif(f_state[.,1 4 5 7]~veg,rcw[.,4].==1);

@ Initialize matrix of cells that are not RCW clusters or in foraging areas @
mostpine = ((f_state[.,16]+f_state[.,20])./f_state[.,6].> 0.5);
non_clus = f_state[.,1 7]~veg~mostpine;

@ If no clusters are found, zero out vectors that contribute to compartment
     rankings @
if clus==miss(0,0);
  nest = 0*comp_all;
  forage = 0*comp_all;
  goto meandens;
endif;

@ NESTING HABITAT @
rcw_list = clus[.,1];
clusmat = (clus[.,4].==comp_all');
```

```
  @ COMPCLUS indicates whether each compartment contains cluster(s) @
  compclus = (sumc(clusmat).>0);

  @ Distributes vegetation density to cluster-compartment combinations @
  vegdist = clusmat.*clus[.,5];

  @ Find the smallest >0 veg density in clusters among all compartments @
  minveg = substute(vegdist,vegdist.==0,1000000);  @ Make 0 values big @
  minveg = selif(vegdist',compclus);
  minveg = minc(minveg');             @ Min veg density by compartment @
  minveg = substute(minveg,minveg.==1000000,0);    @ Restore 0 values @
  minveg = minc(minveg);                       @ Min overall veg value @

  @ Find the maximum veg density in clusters by compartment.  Compartments
      lacking clusters are assigned the MINVEG value. @
  nest = maxc(vegdist);
  nest = substute(nest,1-compclus,minveg);

  @ FORAGING HABITAT @
  @ Cell dimensions @
  h = sqrt(fullsize/(2*sqrt(3)));
  w = h*sqrt(3)/3;
  v = 2*w;
  tol = 0.01;
  @ Distance matrix for RCW clusters @
  dist = selif(hexdist,rcw[.,4].==1);
  @ Identify cells in 48-cell neighborhood, exclude focal cell @
  ring = (dist.>0).*(dist.<(6*h)*(1+tol)) +
         (dist.>4*(v+w)*(1+tol)).*(dist.<(8*h)*(1-tol));
  @ Indicator vector of membership in one or more foraging neighborhoods @
  veges = (sumc(ring).>0);
  forveg = selif(f_state[.,7]~veg,veges);
  comps = unique(forveg[.,1],1);
  compdum = dummy(forveg[.,1],comps);
  compdum = compdum[.,1:cols(compdum)-1];
  formeans = (compdum'*forveg[.,2])./sumc(compdum);
  minveg = minc(formeans);

  @ Assign mean foraging vegetation values to all refuge compartments @
  @   Use MINVEG value if compartment contains no foraging area @
  compclus = (comps.==comp_all');
  comps = sumc(compclus);
  forage = sumc(compclus.*formeans);
```

```
  forage = substute(forage,1-comps,minveg);

  @ RCW_LIST now contains all RCW nesting and foraging cells @
  rcw_list = rcw_list | selif(f_state[.,1],veges);
  rcw_list = unique(rcw_list,1);

  @ NON-CLUSTER HABITAT @
  @ Remove cluster and foraging hexagons from forest-wide list @
  rcw_list = unique(rcw_list,1);
  pullmat = sumc(rcw_list.==f_state[.,1]');
  non_clus = delif(non_clus,pullmat);

  if non_clus==miss(0,0);
    other = 0*comp_all;
    goto mndens2;
  endif;

  meandens:
  @ Find mean vegetation density in all other pine-containing hexagons @
  other = selif(non_clus,non_clus[.,4]);
  other = meanc((other[.,2].==comp_all').*other[.,3]);

  mndens2:
  @ Find average density over cell types @
  combine = comp_all ~ (nest+forage+other)/3;

  @ Sort the results, then extract top N rows @
  sorted = rev(sortc(combine,2));
  top_n = sorted[1:n,.];

  @ Update age-at-last-burn variable for all non-P1 stands in these comps @
  pullmat = sumc(top_n[.,1].==f_state[.,7]');
  f_out[.,8] = substute(f_state[.,8],
                         pullmat.*(f_state[.,17].>=16),f_state[.,17]);
  f_out[.,9] = substute(f_state[.,9],
                         pullmat.*(f_state[.,21].>=16),f_state[.,21]);

  getout:
  retp(f_out,top_n);

endp;
```

```
proc (1) = thinning(f_state,f_attr,comps);
  /****************************************************************************
    Selects stands for thinning for a chosen set of compartments.  Procedure
      accepts as input F_STATE, the current state of the forest hexagon grid,
      F_ATTR, the matrix of permanent hexagon attributes, and COMPS, a vector
      of compartment selections.  Procedure returns F_OUT, a copy of F_STATE
      but with basal area, age of last thinning, and canopy closure values
      updated appropriately for certain pine stands.
    Globals:
      BA_THIN, BA_CEIL, THIN_AGE
      OUT800, GRIDSIZE
      FH1_AGE, FH1_BA1, FH1_BA2, FH1_AREA
  ****************************************************************************/
  local f_out,owner,amt800,fullsize,pull,cand,qual1,qual2,rcw,clus,distclus,
        mindist,pullamt,amt,i,ring_xy,thinpull,
        sel1,sel2,ringarea,hab,tothab,enuf800,testhab,cand1,ratio,
        h,w,v,tol,idx;

  f_out = f_state;

  @ If no compartments are specified, leave procedure @
  if comps==0;
    goto nocand;
  endif;

  @ Ownership matrix:  HEX_ID, 4 ownership proportions in each of 4 buffers @
  owner = f_attr[.,1 10:25];

  @ Estimated proportion (based on ownership and estimated provision)
      of foraging habitat on non-refuge land in circular buffers @
  amt800 = owner[.,7:9]*out800;

  @ Full hexagon size @;
  fullsize = maxc(f_state[.,6]);

  @ Cell dimensions @
  h = sqrt(fullsize/(2*sqrt(3)));
  w = h*sqrt(3)/3;
  v = 2*w;
  tol = 0.01;

  @ Create candidate list of stands from selected compartments @
    @ columns in CAND:  1-6   HEX_ID, ROW, COL, X, Y, INSIDE
                       7-11  AGE_THIN, AREA, AGE, BA, CC for stand 1
                      12-16  AGE_THIN, AREA, AGE, BA, CC for stand 2  @
  pull = sumc(comps.==(f_state[.,7]'));
  cand = selif((f_state[.,1:6 10 16:19 11 20:23]),pull);

  @ Eliminate from list those cells for which neither stand qualifies for
      possible thinning                                                  @
  qual1 = (cand[.,8].>0).*(cand[.,9].>=thin_age).*(cand[.,10].>=ba_ceil);
  qual2 = (cand[.,13].>0).*(cand[.,14].>=thin_age).*(cand[.,15].>=ba_ceil);
  cand = cand ~ qual1 ~ qual2;
  cand = selif(cand,(qual1 .or qual2));

  @ No thinning candidates left --> exit procedure @
  if cand==miss(0,0);
    goto nocand;
  endif;

  @ Attach a column indicator for the RCW cluster evaluation step @
  cand = cand ~ ones(rows(cand),1);

  @ RCW matrix:  HEX_ID, X, Y, Occurrence of cluster, Activity of cluster @
  rcw = f_state[.,1 4 5] ~ (f_state[.,25].>0) ~ f_state[.,24];

  @ Find all cells containing an RCW cluster @
  clus = selif(f_state[.,1:5],rcw[.,4].==1);

  @ Of candidate cells left, locate all RCW clusters within 1000m of cells @
  @   (to narrow the number of RCW foraging habitats that have to be searched)@
  if clus/=miss(0,0);
    @ At least one cluster is present on refuge @

    @ Calculate distances between clusters in CLUS and cells in CAND @
    @   Distance matrix @
    distclus = sqrt( (cand[.,4]-clus[.,4]')^2 +
                     (cand[.,5]-clus[.,5]')^2 );

    @ Distance of each RCW cluster to nearest candidate cell @
    mindist = minc(distclus);

    @ Remove clusters with >1000m minimum distance @
    clus = delif(clus,mindist.>1000);

    @ Check hexagons for sufficient foraging habitat in 800-m radius @
```

```
  if clus/=miss(0,0);
    @ At least one cluster falls within 1000m of a candidate cell @

    @   Pull corresponding values from AMT800 @
    pullamt = sumc(clus[.,1].==(f_state[.,1]'));
    amt = selif(amt800,pullamt);

    @   Search over all the hexagons @
    i = 0;
    do while (i<rows(clus));
      i = i+1;

      @ Find row index of this cell in F_STATE @
      idx = maxindc(clus[i,1].==f_state[.,1]);

      @ Distance vector for this cluster @
      distclus = hexdist[idx,.];

      @ Identify cells in 48-cell neighborhood, include focal cell @
      ring_xy = (distclus.<(6*h)*(1+tol)) +
                (distclus.>4*(v+w)*(1+tol)).*(distclus.<(8*h)*(1-tol));
      ring_xy = ring_xy';

      @ Pull out both pine components for this ring @
      sel1 = selif(f_state[.,1 16 17 18],ring_xy);
      sel2 = selif(f_state[.,1 20 21 22],ring_xy);

      @ Does this foraging area contain any of the candidate thinning cells?@
      thinpull = sumc(sel1[.,1].==(cand[.,1]'));
      if sumc(thinpull)==0;
        @ If not, go to next cluster @
        continue;
      endif;

      @ Calculate area of habitat ring @
      ringarea = sumc(selif(f_state[.,6],ring_xy));

      @ Calculate suitable habitat in ring @
      hab = (sel1[.,3].>=fh1_age).*(sel1[.,4].>=fh1_ba1).*sel1[.,2] +
            (sel2[.,3].>=fh1_age).*(sel2[.,4].>=fh1_ba1).*sel2[.,2];
      tothab = sumc(hab);

      @ Test for sufficiency of habitat w.r.t. location of hexagon @
      enuf800 = 1;
      if rows(sel1)==49 and ringarea/(49*fullsize)>=0.9999;
        @ All habitat in ring entirely on refuge @

        if tothab<fh1_area;  @ Not enough habitat of this type @
          enuf800 = 0;
        endif;

      else;
        @ Some habitat located off-site @

        if tothab<fh1_area;  @ Not enough habitat, check off-refuge @
          @ Estimate of total habitat found on and off-refuge @
          testhab = tothab + 49*fullsize*amt[i];
          if testhab<fh1_area;  @ Still not enough habitat @
            enuf800 = 0;
          endif;
        endif;

      endif;      @ 800-m habitat sufficiency test completed @

      @ If insufficient 800-m foraging habitat, set thinning indicator to 0 @
      if enuf800==0;
        cand[.,19] = recode(cand[.,19],thinpull,0);
      endif;

    endo;     @ Finished looping over all hexagons @

  endif;

endif;

@ At least some hexagons may be thinned @
if sumc(cand[.,19])/=0;

  cand = selif(cand,cand[.,19].==1);

  @ Identify cells in F_STATE with eligible thinnings in each stand @
  if sumc(cand[.,17])>0;
    @ Stand 1 is eligible for thinning in at least 1 hexagon @

    cand1 = selif(cand,cand[.,17]);
    pull = sumc((cand1[.,1].==(f_state[.,1]')));
```

Appendix C.12. Spatially-explicit Refuge simulation model. Continued.

```
      @ Set age at last thin=current age @
      f_out[.,10] = substute(f_state[.,10],pull,f_state[.,17]);

      @ Set basal area to BA_THIN value @
      f_out[.,18] = substute(f_state[.,18],pull,ba_thin);

      @ Open canopy in proportion to amount thinned @
      ratio = ba_thin./f_state[.,18];
      f_out[.,19] = substute(f_state[.,19],pull,ratio.*f_state[.,19]);

    elseif sumc(cand[.,18])>0;
      @ Stand 2 is eligible for thinning in at least 1 hexagon @

      cand1 = selif(cand,cand[.,18]);
      pull = sumc((cand1[.,1].==(f_state[.,1]')));

      @ Set age at last thin=current age @
      f_out[.,11] = substute(f_state[.,11],pull,f_state[.,21]);

      @ Set basal area to BA_THIN value @
      f_out[.,22] = substute(f_state[.,22],pull,ba_thin);

      @ Open canopy in proportion to amount thinned @
      ratio = ba_thin./f_state[.,22];
      f_out[.,23] = substute(f_state[.,23],pull,ratio.*f_state[.,23]);

    endif;

  endif;

  nocand:

  retp(f_out);

endp;



proc (1) = undstory(f_state,f_attr);
  /**************************************************************************
    Calculate understory vegetation density in pine stands given current pine
      overstory state and fire history.  Use Conroy et al. model to estimate
      vegetation density using basal area and canopy closure values from
    F_STATE, and using slope position values from F_ATTR.  Use Masters et al.
    data to reduce vegetation amounts by number of years since burning.
    Procedure accepts as input F_STATE, the current state of the forest
    hexagon grid, and F_ATTR, the matrix of permanent attributes for F_STATE.
    The program creates VEG_OUT, a vector of estimated understory vegetation
    densities.
  Globals:
    BURNPARM, V1...V6
**************************************************************************/
local veg_out,in_codes,out_code,slopepos,match,areas,ages,bas,ccs,burnages,
      ba_part,burnrate,tslb;

@ Re-assign slope position codes to match Conroy's @
   @ 1 = bottom   ==> floodplain-terrace = 6              @
   @ 2 = foot     ==> footslope/toeslope = (4+5)/2 = 4.5  @
   @ 3 = mid      ==> backslope          = 2              @
   @ 4 = shoulder ==> shoulder           = 1              @
   @ 5 = top      ==> summit             = 3              @
in_codes = {1,2,3,4,5};
out_code = {6,4.5,2,1,3};
match = (f_attr[.,3].==in_codes');
slopepos = recode(f_attr[.,3],match,out_code);

@ Extract stand information from F_STATE @
areas = f_state[.,16 20];
ages = f_state[.,17 21];
bas = f_state[.,18 22];
ccs = f_state[.,19 23];
burnages = f_state[.,8 9];

@ Compute veg density @
ba_part = (areas.>0).*v2./bas;
veg_out = v1 + ba_part + v3*ccs + v4*(slopepos^2) + v5*ccs.*slopepos
             + v6*bas.*slopepos;
  @ Convert missing values to 0 @
veg_out = missrv(veg_out,0);
  @ Change negative values to 0 @
veg_out = substute(veg_out,(veg_out.<0),0);

@ Compute time since last burn @
burnrate = 0 | burnparm | 1;     @ Veg rates for 0, 1, 2, 3, 4+ years TSB @
in_codes = in_codes[1:4]-1;      @ Categories for years since burn @
tslb = ages-burnages;            @ Time since last burn @
```

```
    @ Replace times since burn with veg reduction rates for stand 1 @
  match = (tslb[.,1].==in_codes');
  tslb[.,1] = code(match,burnrate);
    @ Replace times since burn with veg reduction rates for stand 2 @
  match = (tslb[.,2].==in_codes');
  tslb[.,2] = code(match,burnrate);

  @ Multiply veg density values by burn reduction values @
  veg_out = veg_out.*tslb;

  @ Weighted average of vegetation density @
  veg_out = sumc((veg_out.*areas)') ./ sumc(areas');
    @ Change missing values to 0 @
  veg_out = missrv(veg_out,0);

  retp(veg_out);

endp;



proc (1) = for_grow(f_state,f_attr);
  /***************************************************************************
    Grow forest cover to next time period.  Use Bailey-Ware model to project
      basal area in response to current age, current basal area, site index,
      and age at last thinning.  Use the Valverde-Silvertown model to project
      degree of canopy closure.  Procedure accepts as input F_STATE, the
      current state of the forest hexagon grid, and F_ATTR, the matrix of
      permanent attributes for F_STATE.  The program creates F_OUT, a copy
      of F_STATE, but reflecting projected basal area and canopy cover for
      pine stands in the next year.
    Globals:
      THININDX, B1, B2, B3, AGE0, BA0, BA_MAX, CCRATE
  ***************************************************************************/
  local x0,si_medn,areas,ages,bas,ccs,agethins,nxtages,tooyoung,unthinned,
        thinned,arat,ba1,part2,prod,ba2,nxtbas,nxtccs,f_out;

  f_out = f_state;
  x0 = (thinindx/=0)*(1-thinindx);       @ thinning index @
  si_medn = f_attr[.,2]/3.280839895;     @ Median of site index (m) for cell @

  areas = f_state[.,16 20];
  ages = f_state[.,17 21];
```

```
  bas = f_state[.,18 22];
  ccs = f_state[.,19 23];
  agethins = f_state[.,10 11];

  nxtages = (areas.>0).*(ages+1);

  @ Indicator matrices for different BA computation @
  tooyoung = (nxtages.<=age0 .and nxtages.>0);  @ Below age threshold @
  unthinned = (1-tooyoung).*(agethins.==0);     @ Not previously thinned @
  thinned = (agethins.>0);                      @ Previously thinned @

  @ Computation for previously unthinned stands @
  arat = (nxtages.>0).*age0./nxtages;
  ba1 = ba0^arat .* exp( b1*(1-arat) + b3*si_medn.*(1-arat) );
  ba1 = missrv(ba1,0);

  @ Computation for previously thinned stands @
  arat = (nxtages.>0).*ages./nxtages;
  part2 = (agethins.>0).*(nxtages.>0)*(b2*x0)./(agethins.*nxtages);
  prod = exp(b1*(1-arat) + (1./nxtages-1./ages).*part2 + b3*si_medn.*(1-arat));
  ba2 = (bas^(arat)).*prod;
  ba2 = missrv(ba2,0);

  @ Combine BA computations @
  nxtbas = tooyoung*ba0 + unthinned.*ba1 + thinned.*ba2;

  @ Enforce maximum BA value @
  nxtbas = substute(nxtbas,nxtbas.>=ba_max,ba_max);

  @ Replace BA in F_STATE matrix, non-pine stands get BA=0 @
  f_out[.,18 22] = (areas.>0).*nxtbas;

  @ Calculate closure in canopy cover @
  nxtccs = 100 - (100-ccs)*ccrate;

  @ Replace CC in F_STATE matrix, non-pine stands get CC=0 @
  f_out[.,19 23] = (areas.>0).*nxtccs;

  @ Replace ages in F_STATE matrix, non-pine stands get age=0 @
  f_out[.,17 21] = (areas.>0).*nxtages;

  retp(f_out);
```

```
endp;



proc (1) = recrclus(f_state,f_attr,rcwquota);
  /***************************************************************************
    Establish a target number of RCW recruitment clusters.  Procedure accepts
      as input F_STATE, the current state of the of the forest hexagon grid,
      F_ATTR, the matrix of permanent attributes for F_STATE, and RCWQUOTA,
      the current quota for recruitment clusters.  The program creates F_OUT,
      a copy of F_STATE but reflecting the new RCW clusters established,
      if any.  Global variables used by the procedure are:
        CLUSGROW, GRIDSIZE,
        OUT400, OUT800,
        RC_AGE1, RC_AGE2, RC_AMT1, RC_AMT2,
        FH2_AGE, FH2_BA1, FH2_BA2, FH2_AREA,
        FH1_AGE, FH1_BA1, FH1_BA2, FH1_AREA,
        CLUS_D1, CLUS_D2A, CLUS_D2B, CLUS_INC
  ***************************************************************************/

  local rcw,rcwtotal,rcwactiv,pine1,pine2,fullsize,hexes,owner,
        amt400,amt800,rcw_need,rcw_left,cand,cand_n,age1hab,age2hab,
        cand1,enf_400,i,col,row,ring_xy,index_xy,pullmat,sel1,sel2,
        ringarea,hab,tothab,testhab,recr,calcdist,rcw_act,rcw_any,
        candonly,nearclus,dist_act,dist_any,dist_max,nearsel,
        cand800,pullamt,amt,winners,winner,win_xy,f_out,idx,
        h,w,v,tol,distclus,timer;

  timer=date;

  f_out = f_state;

  @ RCW matrix:  HEX_ID, Occurrence of cluster, Activity of cluster @
  rcw = f_state[.,1] ~ (f_state[.,25].>0) ~ f_state[.,24];
  rcwtotal = sumc(rcw[.,2]);
  rcwactiv = sumc(rcw[.,3]);

  @ PINE matrices for stands 1 and 2:  HEX_ID, AREA, AGE, BA, and CC @
  pine1 = f_state[.,1 16:19];
  pine2 = f_state[.,1 20:23];

  @ Full hexagon size @;
  fullsize = maxc(f_state[.,6]);
```

```
@ Rows in F_STATE matrix @;
hexes = rows(f_state);

@ Cell dimensions @
h = sqrt(fullsize/(2*sqrt(3)));
w = h*sqrt(3)/3;
v = 2*w;
tol = 0.01;

@ Ownership matrix:  HEX_ID, 4 ownership proportions in each of 4 buffers @
owner = f_attr[.,1 10:25];

@ Estimated proportion (based on ownership and estimated provision)
    of foraging habitat on non-refuge land in circular buffers @
amt400 = owner[.,3:5]*out400;
amt800 = owner[.,7:9]*out800;

@ Calculate number of clusters needed and initialize number left to fill @
rcw_need = rcwquota - (rcwtotal-rcwactiv);
rcw_left = round(rcw_need);

@ Accumulator of selected recruitment clusters @
recr = {};

if rcw_left>0;

  @ From initial list (F_STATE[26]), eliminate hexagons already as clusters @
  cand = f_state[.,1 26];
  cand[.,2] = cand[.,2].*(rcw[.,2].==0);
  cand_n = sumc(cand[.,2]);

  @ Exit if no candidate cells left @
  if cand_n==0;
    goto empty;
  endif;

  @ Eliminate hexagons with insufficient nesting habitat @
  @   Calculate amount of habitat at age RC_AGE1 or older @
  age1hab = pine1[.,1] ~ ( (pine1[.,3].>=rc_age1).*pine1[.,2] +
                           (pine2[.,3].>=rc_age1).*pine2[.,2] );

  @   Calculate amount of habitat at age RC_AGE2 or older @
  age2hab = pine1[.,1] ~ ( (pine1[.,3].>=rc_age2).*pine1[.,2] +
```

Appendix C.12. Spatially-explicit Refuge simulation model. Continued.

```
                             (pine2[.,3].>=rc_age2).*pine2[.,2] );

    @   Do these amounts exceed threshold sizes for hexagon? @
    cand[.,2] = cand[.,2].*(age1hab[.,2]/fullsize .>= rc_amt1);
    cand[.,2] = cand[.,2].*(age2hab[.,2]/fullsize .>= rc_amt2);
    cand_n = sumc(cand[.,2]);

    @ Exit if no candidate cells left @
    if cand_n==0;
      goto empty;
    endif;

    @ Eliminate hexagons with insufficient foraging habitat in 400-m radius @
    cand1 = cand;   @ Preserve original candidate list @
    enf_400 = 1;    @ Indicator of stringent (1) or relaxed foraging standard @


/**************************  Test removal of 400-m loop *********************


    @   Search over all the hexagons @
    i = 0;
    do while (i<hexes);
      i = i+1;

      if (cand[i,2]==1);     @ Find those that are still candidates @


        @ Distance vector for this cluster @
        distclus = hexdist[i,.];

        @ Identify cells in 12-cell neighborhood, include focal cell @
        ring_xy = (distclus.<2*(v+w)*(1+tol));
        pullmat = ring_xy';

/*****
        @ Get row and column indices of focal hexagon @
        col = f_out[i,3];
        row = f_out[i,2];

        @ Pull out the ring of 12-nearest neighbor hexagons for this cell @
        ring_xy = ring(col,row,gridsize,12);
```

```
        @ Convert cell addresses to indices @
        index_xy = gridindx((ring_xy|(col~row)),gridsize);

        @ Match ring indices to indices in full data matrix @
        pullmat = sumc(index_xy.==(f_out[.,1]'));
*****/

        @ Pull out both pine components for this ring @
        sel1 = selif(pine1,pullmat);
        sel2 = selif(pine2,pullmat);

        @ Calculate area of habitat ring @
        ringarea = sumc(selif(f_out[.,6],pullmat));

        @ Calculate suitable habitat in ring @
        hab = (sel1[.,3].>=fh2_age).*(sel1[.,4].>=fh2_ba1).*
                    (sel1[.,4].<=fh2_ba2).*sel1[.,2] +
              (sel2[.,3].>=fh2_age).*(sel2[.,4].>=fh2_ba1).*
                    (sel2[.,4].<=fh2_ba2).*sel2[.,2];
        tothab = sumc(hab);

        @ Test for sufficiency of habitat w.r.t. location of hex @
        if rows(sel1)==13 and ringarea/(13*fullsize)>=0.9999;
          @ All habitat in ring entirely on refuge @

          if tothab<fh2_area;  @ Not enough habitat of this type @
            cand[i,2] = 0;
          endif;

        else;
          @ Some habitat located off-site @

          if tothab<fh2_area;  @ Not enough habitat, check off-refuge @
            @ Estimate of total habitat found on and off-refuge @
            testhab = tothab + 13*fullsize*amt400[i];
            if testhab<fh2_area;  @ Still not enough habitat @
              cand[i,2] = 0;
            endif;
          endif;

        endif;      @ Habitat sufficiency test completed @

      endif;     @ End processing of this candidate hexagon @
```

Appendix C.12. Spatially-explicit Refuge simulation model. Continued.

```
    endo;     @ Finished looping over all hexagons @



**************************  Test removal of 400-m loop *********************/



    if sumc(cand[.,2])==0;
      @ No hexagons met this foraging standard--relax standard & proceed @
      enf_400 = 0;     @ Stringent standard not enforced @
      cand = cand1;    @ Bring back original list of candidate hexagons @
    endif;

    @ Switch for controlling calculation of distances @
    calcdist = 1;

    @ Matrices of X-Y locations for active and all clusters @
    rcw_act = selif(f_out[.,1 4 5],rcw[.,3].==1);
    rcw_any = selif(f_out[.,1 4 5],rcw[.,2].==1);

    @ Repeat hexagon selection below until quota is met @
    do while rcw_left>0;

      @ First, calculate distances between candidate cells and both active
              and all clusters @
      if calcdist==1;

        calcdist = 0;   @ Don't calculate this again unless switch is reset @

        @ Extract cells still in the candidate list @
        candonly = selif(f_out[.,1 4 5],cand[.,2].==1);
        if candonly==miss(0,0);
          goto empty;
        endif;

        @ X-Y locations of cells still in the candidate list @
        nearclus = candonly[.,1] ~ ones(rows(candonly),1);

        @ Distances from candidate cells to active clusters @
        dist_act = sqrt( (rcw_act[.,2]-candonly[.,2]')^2 +
                         (rcw_act[.,3]-candonly[.,3]')^2 );
```

```
  nearclus = nearclus ~ minc(dist_act);   @ Append nearest neighbors @

  @ Distances from candidate cells to all clusters @
  dist_any = sqrt( (rcw_any[.,2]-candonly[.,2]')^2 +
                   (rcw_any[.,3]-candonly[.,3]')^2 );
  nearclus = nearclus ~ minc(dist_any);   @ Append nearest neighbors @

endif;

@ Eliminate hexagons too close to any other cluster @
nearclus[.,2] = nearclus[.,2].*(nearclus[.,4].>=clus_d1);

@ Are remaining hexagons sufficiently far from other clusters? @
if sumc(nearclus[.,2])==0;
  @ No hexagons met criterion (all were too close to clusters) @

  if enf_400==0;
    @ Already using lenient 400-m foraging criterion--exit loop @
    break;

  else;
    @ Use lenient 400-m foraging criterion and try again @
    enf_400 = 0;
    cand = cand1;
    calcdist = 1;
    continue;

  endif;

endif;

@ Check hexagons for proximity to active clusters,
    DIST_MAX is current distance threshold, initialized at CLUS_D2A    @
dist_max = clus_d2a;

@ Return here if DIST_MAX is too small @
dist:

@ NEARSEL is selection of viable hexagons at current distance threshold @
nearsel = nearclus;
nearsel[.,2] = nearsel[.,2].*(nearsel[.,3].<dist_max);

if sumc(nearsel[.,2])==0;
```

Appendix C.12. Spatially-explicit Refuge simulation model. Continued.

```
        @ No hexagons met criterion (all were too far from active clusters) @

        if dist_max>clus_d2b or abs(dist_max-clus_d2b)<0.0001;
          @ At limit of distance test @

          if enf_400==0;
            @ Already using lenient 400-m foraging criterion--exit loop @
            break;

          else;
            @ Use lenient 400-m foraging criterion and try again @
            enf_400 = 0;
            cand = cand1;
            calcdist = 1;
            continue;

          endif;

        else;
          @ Not at distance limit, bump up to next increment @
          dist_max = dist_max + clus_inc;
          goto dist;   @ Try next distance increment @
        endif;

      endif;

      @ Check hexagons for sufficient foraging habitat in 800-m radius @
      @   CAND800 holds IDs of cells passing current maximum distance crit @
      cand800 = selif(nearsel[.,1 2],nearsel[.,2].==1);


/******************  Test removal of 800-m loop ********************


      @   Append row & column addresses to CAND800 @
      cand800 = cand800 ~ indxgrid(cand800[.,1],gridsize);

      @   Pull corresponding values from AMT800 @
      pullamt = sumc(cand800[.,1].==(f_out[.,1]'));
      amt = selif(amt800,pullamt);

      @   Search over all the hexagons @
      i = 0;
```

```
      do while (i<rows(cand800));
        i = i+1;


        @ Find row index of this cell in F_STATE @
        idx = maxindc(cand800[i,1].==f_state[.,1]);

        @ Distance vector for this cluster @
        distclus = hexdist[idx,.];

        @ Identify cells in 48-cell neighborhood, include focal cell @
        ring_xy = (distclus.<(6*h)*(1+tol)) +
                  (distclus.>4*(v+w)*(1+tol)).*(distclus.<(8*h)*(1-tol));
        pullmat = ring_xy';

/*****
        @ Get row and column indices of focal hexagon @
        col = cand800[i,3];
        row = cand800[i,4];

        @ Pull out the ring of 48-nearest neighbor hexagons for this cell @
        ring_xy = ring(col,row,gridsize,48);

        @ Convert cell addresses to indices @
        index_xy = gridindx((ring_xy|(col~row)),gridsize);

        @ Match ring indices to indices in full data matrix @
        pullmat = sumc(index_xy.==(f_out[.,1]'));
*****/

        @ Pull out both pine components for this ring @
        sel1 = selif(pine1,pullmat);
        sel2 = selif(pine2,pullmat);

        @ Calculate area of habitat ring @
        ringarea = sumc(selif(f_out[.,6],pullmat));

        @ Calculate suitable habitat in ring @
        hab = (sel1[.,3].>=fh1_age).*(sel1[.,4].>=fh1_ba1).*
                    (sel1[.,4].<=fh1_ba2).*sel1[.,2] +
              (sel2[.,3].>=fh1_age).*(sel2[.,4].>=fh1_ba1).*
                    (sel2[.,4].<=fh1_ba2).*sel2[.,2];
        tothab = sumc(hab);
```

Appendix C.12. Spatially-explicit Refuge simulation model. Continued.

```
        @ Test for sufficiency of habitat w.r.t. location of hexagon @
        if rows(sel1)==49 and ringarea/(49*fullsize)>=0.9999;
          @ All habitat in ring entirely on refuge @

          if tothab<fh1_area;  @ Not enough habitat of this type @
            cand800[i,2] = 0;
          endif;

        else;
          @ Some habitat located off-site @

          if tothab<fh1_area;  @ Not enough habitat, check off-refuge @
            @ Estimate of total habitat found on and off-refuge @
            testhab = tothab + 49*fullsize*amt[i];
            if testhab<fh1_area;  @ Still not enough habitat @
              cand800[i,2] = 0;
            endif;
          endif;

        endif;      @ 800-m habitat sufficiency test completed @

      endo;     @ Finished looping over all hexagons @



******************  Test removal of 800-m loop ********************/



      if sumc(cand800[.,2])==0;
        @ No hexagons met 800-m foraging criterion @

        if dist_max>clus_d2b or abs(dist_max-clus_d2b)<0.0001;
          @ At limit of distance test @

          if enf_400==0;
            @ Already using lenient 400-m foraging criterion--exit loop @
            break;

          else;
            @ Use lenient 400-m foraging criterion and try again @
            enf_400 = 0;
            cand = cand1;
```

```
      calcdist = 1;
      continue;

    endif;

  else;

    @ Not at distance limit, bump up to next increment @
    dist_max = dist_max + clus_inc;
    goto dist;   @ Try next distance increment @

  endif;

endif;

@ Extract indices of cells in CAND800 passing all criteria @
winners = selif(cand800[.,1],cand800[.,2].==1);

@ Find these hexagons in PINE1 and PINE2 matrices @
pullmat = sumc(winners.==(pine1[.,1]'));
sel1 = selif(pine1,pullmat);
sel2 = selif(pine2,pullmat);

@ Calculate area-weighted average of age of pine stands @
winners = winners ~
   (sel1[.,2].*sel1[.,3]+sel2[.,2].*sel2[.,3]) ./ (sel1[.,2]+sel2[.,2]);

@ Find hexagon index of oldest stand meeting all criteria @
winner = winners[maxindc(winners[.,2]),1];

@ Extract row, column indices of WINNER cell, add to RECR @
win_xy = selif(f_out[.,1 4 5],f_out[.,1].==winner);
recr = recr | win_xy;

@ Change cluster status of RCW from absent (0) to present (1) @
rcw[.,2] = rcw[.,2] + (rcw[.,1].==winner);

@ Change cluster status in F_STATE from absent (0) to present (1) @
f_out[.,25] = f_out[.,25] + (rcw[.,1].==winner);

@ Reduce quota by 1 @
rcw_left = rcw_left-1;
```

Appendix C.12. Spatially-explicit Refuge simulation model. Continued.

```
      @ More clusters to establish? @
      if rcw_left>0;
        @ Remove WINNER hexagon from CAND1 (lenient) and CAND (stringent) @

        cand1[.,2] = cand1[.,2].*(cand1[.,1]./=winner);
        cand[.,2] = cand[.,2].*(cand[.,1]./=winner);

        @ Add WINNER hexagon to RCW_ANY list @
        rcw_any = rcw_any | win_xy;

        @ Force a recalculation of distances @
        calcdist = 1;

      endif;

    endo;    @ Find next cluster @

  endif;   @ No more clusters needed @

  empty:

  retp(f_out);

endp;



proc (1) = regencut(f_state,comps);
  /***************************************************************************
    Determines location and quantity of regeneration cuts.  Procedure accepts
      as input F_STATE, the current state of the forest hexagon grid, and
      COMPS, the column vector of compartments to be treated at this time
      period.  The program creates F_OUT, a copy of F_STATE that reflects
      the size and locations of regeneration cuts.  Global variables used by
      the procedure are:
        DECTABLE, DEC_PARM,
        GRIDSIZE,
        WT_AREA, WT_DIST,
        CUTLIMIT, BAO
  ***************************************************************************/

  local rcw,pine1,pine2,uh,uh_tot,p1,p2,p3,p4,p1_tot,p2_tot,p3_tot,p4_tot,
        for_tot,statevec,dec,quota,quota1,unfilled,pull,cand,pullmat,
        p1c,p2c,p3c,p4c,clus,row_clus,clus_all,i,col,row,ring_xy,index_xy,
        p1cell,p1_all,row_p1,distclus,dcols,mean_nn,distvec,sorted,
        pc1,pc2,maxes,mins,range,areastd,areastd0,nndstd,avg_ad,wt_d,
        regen,cutpatch,toptype,avgcol,topcell,j,seedgrp,tt,quo,stand,
        agecol,areacol,typecol,ptype,cutleft,s1lim,s2lim,seed_all,
        pullage,totalcut,f_out,h,w,v,tol,dist;

  f_out = f_state;

  @ RCW matrix:  HEX_ID, Occurrence of cluster, Activity of cluster @
  rcw = f_state[.,1] ~ (f_state[.,25].>0) ~ f_state[.,24];

  @ PINE matrices for stands 1 and 2:  HEX_ID, AREA, AGE, BA, and CC @
  pine1 = f_state[.,1 16:19];
  pine2 = f_state[.,1 20:23];

  @ Upland hardwood area @
  uh = f_state[.,1 13];

  @ Compute proportions of forest in pine age classes and in upland hardwood @
  uh_tot = sumc(uh[.,2]);
  p1 = pine1[.,2].*(pine1[.,3].<16) + pine2[.,2].*(pine2[.,3].<16);
  p2 = pine1[.,2].*(pine1[.,3].>=16).*(pine1[.,3].<40) +
       pine2[.,2].*(pine2[.,3].>=16).*(pine2[.,3].<40);
  p3 = pine1[.,2].*(pine1[.,3].>=40).*(pine1[.,3].<80) +
       pine2[.,2].*(pine2[.,3].>=40).*(pine2[.,3].<80);
  p4 = pine1[.,2].*(pine1[.,3].>=80) + pine2[.,2].*(pine2[.,3].>=80);
  p1_tot = sumc(p1);
  p2_tot = sumc(p2);
  p3_tot = sumc(p3);
  p4_tot = sumc(p4);
  for_tot = p1_tot+p2_tot+p3_tot+p4_tot+uh_tot;

  @ State vector input to optimal regen decision look-up table @
  statevec = (p1_tot | p2_tot | p3_tot | p4_tot | uh_tot) / for_tot;

  @ Look up optimal decision values for this system state @
  dec = dlookup(statevec,dectable,dec_parm);

  @ Calculate quota to regenerate @
  quota = dec.*(p2_tot | p3_tot | p4_tot);

  @ QUOTA1 is iteratively reduced toward 0 as patches are found and cut @
```

Appendix C.12. Spatially-explicit Refuge simulation model. Continued.

```
quota1 = quota ~ (2|3|4);

@ UNFILLED holds unmet quota after a pine type is exhausted @
unfilled = 0*quota1;

@ Initialize matrix of cells containing regeneration patches @
regen = {};

@ Look for regeneration patches if QUOTA is non-zero @
if sumc(quota)>0 and sumc(comps)>0;

  @ Select hexagons belonging to treated compartments @
  pull = sumc(comps.==(f_state[.,7]'));
  cand = selif((f_state[.,1:6 16 17 20 21]),pull);

  @ Calculate total area of each pine type in each candidate cell @
  p1c = cand[.,7].*(cand[.,8].<16) + cand[.,9].*(cand[.,10].<16);
  p2c = cand[.,7].*(cand[.,8].>=16).*(cand[.,8].<40) +
        cand[.,9].*(cand[.,10].>=16).*(cand[.,10].<40);
  p3c = cand[.,7].*(cand[.,8].>=40).*(cand[.,8].<80) +
        cand[.,9].*(cand[.,10].>=40).*(cand[.,10].<80);
  p4c = cand[.,7].*(cand[.,8].>=80) + cand[.,9].*(cand[.,10].>=80);
  cand = cand~p1c~p2c~p3c~p4c;

  @ Find all cells containing an RCW cluster @
  clus = selif(f_state[.,1:5],rcw[.,2].==1);
  row_clus = rows(clus);

  @ Find all cells adjacent to RCW clusters @
  if clus/=miss(0,0);
    @ At least one cluster is present on refuge @

    @ Cell dimensions @
    h = sqrt(fullsize/(2*sqrt(3)));
    w = h*sqrt(3)/3;
    v = 2*w;
    tol = 0.01;

    @ Distance matrix for RCW clusters @
    dist = selif(hexdist,rcw[.,2].==1);
    @ Identify cells in 6-cell neighborhood, including focal cell @
    ring_xy = (dist.<(2*h)*(1+tol));

    @ Indicator vector of cluster or cluster ring @
    clus_all = (sumc(ring_xy).>0);

    @ Cell ID of cluster or cluster ring @
    clus_all = selif(f_state[.,1],clus_all);

    @ Locate all RCW and RCW-neighbor cells in CAND and remove them @
    pull = sumc(clus_all.==(cand[.,1]'));
    cand = delif(cand,pull);

  endif;

  @ Exit if no candidate cells left @
  if cand==miss(0,0);
    goto nocand;
  endif;

  @ Find all cells containing P1 stands @
  p1cell = selif(f_state[.,1:3],p1.>0);
  row_p1 = rows(p1cell);

  @ Find all cells adjacent to P1 stands @
  if p1cell/=miss(0,0);
    @ At least one P1 stand is present on refuge @

    @ Distance matrix for P1-containing cells @
    dist = selif(hexdist,p1.>0);

    @ Identify cells in 6-cell neighborhood, including focal cell @
    ring_xy = (dist.<(2*h)*(1+tol));

    @ Indicator vector of P1 cell or cell ring @
    p1_all = (sumc(ring_xy).>0);

    @ Cell ID of cluster or cluster ring @
    p1_all = selif(f_state[.,1],p1_all);

    @ Locate all P1 and P1-neighbor cells in CAND and remove them @
    pull = sumc(p1_all.==(cand[.,1]'));
    cand = delif(cand,pull);

  endif;
```

Appendix C.12. Spatially-explicit Refuge simulation model. Continued.

```
@ Exit if no candidate cells left @
if cand==miss(0,0);
  goto nocand;
endif;

@ Remove cells not providing any harvestable pine stands @
cand = delif(cand,(cand[.,12].==0).*(cand[.,13].==0).*(cand[.,14].==0));

@ Exit if no candidate cells left @
if cand==miss(0,0);
  goto nocand;
endif;

@ Calculate distances between clusters in CLUS and cells in CAND @
if clus/=miss(0,0);
  @ At least one cluster is present on refuge @

  @ Distance matrix @
  distclus = sqrt( (clus[.,4]-cand[.,4]')^2 +
                   (clus[.,5]-cand[.,5]')^2 );
  dcols = cols(distclus);         @ Number of cells in CAND @

  @ Find distance of 3 nearest RCW clusters to each cell in CAND @
  mean_nn = {};
  i = 0;
  do while i<dcols;  @ Search over all columns of DISTCLUS @
    i = i+1;

    @ Sorted vector of distances for candidate cell I @
    distvec = distclus[.,i];
    sorted = sortc(distvec,1);

    @ Extract up to 3 elements, calculate mean distance @
    if row_clus<3;
      mean_nn = mean_nn | meanc(sorted);
    else;
      mean_nn = mean_nn | meanc(sorted[1:3]);
    endif;

  endo;

else;
```

```
  @ No clusters are present--set mean distance to -1 @
  mean_nn = -1*ones(rows(cand),1);

endif;

@ Append the mean nearest-neighbor distances to CAND @
cand = cand ~ mean_nn;

@ Identify age class of each of the 2 pine stands in each cell @
pc1 = 2*(cand[.,8].>=16).*(cand[.,8].<40) +
      3*(cand[.,8].>=40).*(cand[.,8].<80) +
      4*(cand[.,8].>=80);
pc2 = 2*(cand[.,10].>=16).*(cand[.,10].<40) +
      3*(cand[.,10].>=40).*(cand[.,10].<80) +
      4*(cand[.,10].>=80);

@ Append age class indicators to CAND @
cand = cand~pc1~pc2;

@ Find range of area in each pine type and range of NN distance @
maxes = maxc(cand[.,12:15]);
mins = minc(cand[.,12:15]);
range = maxes - mins;

@ Standardize areas for each pine type @
areastd = {};
i = 0;
@ Iterate over each pine type @
do while i<3;
  i = i+1;

  @ Selects appropriate column for pine type @
  col = i+11;

  @ Is range element non-zero? @
  if range[i]>0;

    @ Yes, standardization can proceed for this pine type @
    areastd0 = cand[.,col]/range[i];

  else;

    @ No, this area cannot be standardized because range = 0 @
```

Appendix C.12. Spatially-explicit Refuge simulation model. Continued.

```
    @ Does any of this type exist in CAND? @
    if maxes[i]==0;

      @ No!  Set UNFILLED for this type and zero out QUOTA1 @
      areastd0 = cand[.,col];
      unfilled[i,1] = quota1[i,1];
      quota1[i,1] = 0;

    else;

      @ Yes, but all values are the same-->do not standardize area @
      areastd0 = cand[.,col];

    endif;

  endif;

  @ Append standardized column to AREASTD @
  areastd = areastd ~ areastd0;

endo;      @ Go to the next pine type @

@ Standardize nearest-neighbor distances @
wt_d = wt_dist;
if range[4]>0;

  @ Range is nonzero, standardization is OK @
  nndstd = cand[.,15]/range[4];

else;

  @ NN distance cannot be standardized because range = 0 @
  nndstd = cand[.,15];

  if nndstd[1]==-1;

    @ No clusters exist, change distance values to 1 and set weight to 0 @
    nndstd = -1*nndstd;
    wt_d = 0;

  endif;

endif;
```

```
@ Obtain weighted geometric means of scaled areas and scaled distance @
avg_ad = ((areastd^wt_area).*(nndstd^wt_d))^(1/(wt_area+wt_d));

cand = cand~avg_ad;   @ CAND now has 20 columns @

@ Search for patch seed cells until candidate list exhausted or all @
@   quotas filled                                                  @
do until cand==miss(0,0) or quota1[.,1]<=0;

  @ Initialize patch size variable @
  cutpatch = 0;

  @ Find the pine type with highest regeneration quota @
  toptype = sortc(quota1,1);
  toptype = toptype[3,2];

  @ Column indicator of weighted average corresponding to TOPTYPE @
  avgcol = 16+toptype;

  @ Sort candidate list on this column, then pull out row with maximum   @
  @   average (top row)--> This becomes the seed cell                    @
  topcell = rev(sortc(cand,avgcol));
  topcell = topcell[1,.];

  @ Does any of this pine type remain? @
  if topcell[avgcol]==0;

    @ If not, update UNFILLED matrix, zero out QUOTA1,  @
    @    and continue to next seed cell                 @
    unfilled[toptype-1,1] = quota1[toptype-1,1];
    quota1[toptype-1,1] = 0;
    continue;

  endif;

  @ Pull out regeneration acreage from each pine stand of seed cell @
  i = 0;
  @   Interrogate each pine age class @
  do while i<3;
    i = i+1;

    @ Age class indicator @
    j = i+1;
```

Appendix C.12. Spatially-explicit Refuge simulation model. Continued.

```
  @ Does quota for this class exceed 0? @
  if quota1[i,1]>0;
    @ Yes, proceed with harvest @

    @ Does this age class occur in stand 1? @
    if topcell[16]==j;

      @ Yes, regenerate stand 1 @
      quota1[i,1] = quota1[i,1]-topcell[7];     @ Reduce this quota @
      cutpatch = cutpatch + topcell[7];       @ Augment patch total @
      topcell[8] = 0;            @ Reset age for this stand to zero @

    endif;

    @ Does this age class occur in stand 2? @
    if topcell[17]==j;

      @ Yes, regenerate stand 2 @
      quota1[i,1] = quota1[i,1]-topcell[9];     @ Reduce this quota @
      cutpatch = cutpatch + topcell[9];       @ Augment patch total @
      topcell[10] = 0;           @ Reset age for this stand to zero @

    endif;

  endif;

endo;         @ Get next age class @

@ Remove seed cell from candidate list and begin a regen list @
cand = delif(cand,cand[.,1].==topcell[1]);
seedgrp = topcell;

@ Leave loop if no candidates left or quotas satisfied @
if cand==miss(0,0) or quota1[.,1]<=0;
  break;
endif;

@ Find 6 adjacent cells to seed cell @
ring_xy = ring(topcell[3],topcell[2],gridsize,6);
index_xy = gridindx(ring_xy,gridsize);

@ Remove those adjacent cells not in the candidate list @
pull = sumc(index_xy.==(cand[.,1]'));
```

```
index_xy = selif(cand,pull);

@ If no adjacent cells meet criteria, restart loop with a new seed cell @
if index_xy==miss(0,0);
  continue;
endif;

@ Search all cells in the neighborhood of the seed cell @
i = rows(index_xy);
do while i>0;
  i = i-1;

  @ Select ring cell with highest area in TOPTYPE @
  col = toptype+10;            @ Column of INDEX_XY for TOPTYPE @
  tt = toptype-1;            @ TOPTYPE row indicator for QUOTA1 @
  quo = 1;            @ Indicator of non-zero quota for TOPTYPE @
  index_xy = rev(sortc(index_xy,col));

  @ TOPCELL is cell with highest area in TOPTYPE @
  topcell = index_xy[1,.];

  @ Is this type in both pine stands? @
  if topcell[16]==toptype and topcell[17]==toptype;

    @ Yes, find stand with greatest area @
    stand = 1*(topcell[7]>topcell[9]) + 2*(topcell[7]<=topcell[9]);

  elseif topcell[16]==toptype or topcell[17]==toptype;

    @ No, but this type occurs in one of the stands @
    stand = 1*(topcell[16]==toptype) + 2*(topcell[17]==toptype);

  else;

    @ Type is in neither stand--check the other types instead @
    stand = 1;               @ Arbitrarily choose stand 1 to cut first @
    tt = topcell[16]-1;    @ Set row indicator for this type in QUOTA1 @
    @ QUO now holds quota value for this type @
    quo = selif(quota1[.,1],quota1[.,2].==topcell[16]);

  endif;

  @ Has quota for this type been filled? @
```

```
if quo>0;

  @ No--investigate this stand for cutting @
  @ Age and area columns corresponding to STAND @
  agecol = 8*(stand==1) + 10*(stand==2);
  areacol = agecol-1;

  @ Would cutting this stand send patch size beyond its limit? @
  if cutpatch+topcell[areacol]<cutlimit;

    @ No--cut the stand, and reduce quota for this type @
    quota1[tt,1] = quota1[tt,1] - topcell[areacol];
    cutpatch = cutpatch + topcell[areacol];   @ Augment patch total @
    topcell[agecol] = 0;            @ Reset age for this stand to 0 @

  endif;

endif;

@ Should/can the other stand be cut?:          @
@   Identify pine type in other stand, if any @
stand = 3 - stand;
typecol = 16*(stand==1) + 17*(stand==2);
ptype = topcell[typecol];

@   QUO holds quota value for this type @
quo = selif(quota1[.,1],quota1[.,2].==ptype);

@ Has quota for this type been filled? @
if quo>0;

  @ No--investigate this stand for cutting @
  @ Age and area columns corresponding to STAND @
  agecol = 8*(stand==1) + 10*(stand==2);
  areacol = agecol-1;

  @ Would cutting this stand send patch size beyond its limit? @
  if cutpatch+topcell[areacol]<cutlimit;

    @ No--cut the stand, and reduce quota for this type @
    quota1[ptype-1,1] = quota1[ptype-1,1] - topcell[areacol];
    cutpatch = cutpatch + topcell[areacol];   @ Augment patch total @
    topcell[agecol] = 0;            @ Reset age for this stand to 0 @
```

```
    endif;

  endif;

  @ Remove this cell from candidate groups and add to regen list @
  cand = delif(cand,cand[.,1].==topcell[1]);
  index_xy = delif(index_xy,index_xy[.,1].==topcell[1]);
  seedgrp = seedgrp | topcell;

  @ Leave loop if no candidates left or all quotas satisfied @
  if i==0 or cand==miss(0,0) or quota1[.,1]<=0;
    break;
  endif;

  @ Check whether any cell exists that contains a pine stand that  @
  @    would not exceed the patch size limit when cut              @
  @    If there are none (i.e., all stands too big), leave loop    @

  @ Determine the allowable amount left to cut @
  cutleft = cutlimit-cutpatch;

  @ Vectors to indicate existence of stands exceeding CUTLEFT @
  s1lim = (index_xy[.,7].>cutleft) .or (index_xy[.,7].==0);
  s2lim = (index_xy[.,9].>cutleft) .or (index_xy[.,9].==0);

  @ Are all stands in remaining ring cells larger than CUTLEFT? @
  if s1lim.*s2lim==1;
    @ Yes--exit the ring cell loop and search for another seed cell @
    break;
  endif;

endo;       @ Go to next cell in ring @

@ Add regenerated seed cell and neighbor cells to REGEN @
regen = regen | seedgrp;

@ Remove cells from CAND that are adjacent to those just regenerated @
seed_all = {};
i = 0;

@ Consider the seed cell and neighboring regenerated cells @
do while i<rows(seedgrp);
  i = i+1;
```

Appendix C.12. Spatially-explicit Refuge simulation model. Continued.

```
        @ Extract row and column indices of the focal cell @
        col = seedgrp[i,3];
        row = seedgrp[i,2];

        @ Pull out the ring of 6 adjacent hexagons for this cell @
        ring_xy = ring(col,row,gridsize,6);

        @ Convert cell addresses to indices @
        index_xy = gridindx((ring_xy|(col~row)),gridsize);

        @ Append these cell addresses to SEED_ALL @
        seed_all = seed_all | index_xy;

      endo;    @ Get next cell @

      @ Remove duplicate addresses from SEED_ALL @
      seed_all = unique(seed_all,1);

      @ Locate all neighbor cells in CAND and remove them @
      pull = sumc(seed_all.==(cand[.,1]'));
      cand = delif(cand,pull);

      @ Exit if no candidate cells left @
      if cand==miss(0,0);
        goto nocand;
      endif;

    endo;     @ Search for next seed cell @

  endif;

  nocand:
    totalcut = quota-quota1[.,1]-unfilled[.,1];
    unfilled = quota1[.,1].*(quota1[.,1].>0) + unfilled[.,1];

  @ Update F_STATE matrix with age data in REGEN @
  if regen/=miss(0,0);

    @ Identify cells in F_STATE that were regenerated @
    pullmat = (regen[.,1].==(f_state[.,1]'));
    pull = sumc(pullmat);

    @ Set age=0 for regeneration conducted in stand 1 @
    pullage = sumc(pullmat.*regen[.,8]);
    f_out[.,17] = (pull.==0).*f_state[.,17] + (pull.==1).*pullage;

    @ Set age=0 for regeneration conducted in stand 2 @
    pullage = sumc(pullmat.*regen[.,10]);
    f_out[.,21] = (pull.==0).*f_state[.,21] + (pull.==1).*pullage;

  endif;

  @ Age at last thinning variable is also set to 0 @
  f_out[.,10 11] = f_state[.,10 11].*(f_out[.,17 21]./=0);

  @ Basal area is set to BA0 @
  f_out[.,18 22] = f_state[.,18 22].*(f_out[.,17 21]./=0) +
                                 ba0*(f_out[.,17 21].==0);

  @ Canopy cover variable is set to 0 @
  f_out[.,19 23] = f_state[.,19 23].*(f_out[.,17 21]./=0);

  retp(f_out);

endp;
```

Appendix C.13. *Compartment permutations.G*. GAUSS program file randomly searches for 10,000 permutations of the Piedmont National Wildlife Refuge compartment list that meet the criterion for compartment non-adjacency within a management group. Program writes the output file GOODONES, a GAUSS matrix file containing valid permutations.

```
new;

proc (1) = combin(n,k);
  /***************************************************************************
  Computes combinatorial function for integers N and K, returns integer COMBIN.
  ***************************************************************************/
  local combin,diff,x,i;

  if k==0 or k==n;
    combin = 1;
  else;
    diff = n-k;
    if k<diff;
      x = k;
    else;
      x = diff;
    endif;
    combin = 1;
    i = 0;
    do while i<x;
      i = i+1;
      combin = combin*((n-i+1)/i);
    endo;
  endif;
  combin = round(combin);
  retp(combin);

endp;

proc (1) = rnk_kgrp(n,x,sorted);
  /*************************************************************************
  Compute the rank order R of a vector of objects chosen K at a time from N
    total objects.  X is a row vector of size K.  If X is already sorted,
    set SORTED = 1; otherwise, SORTED = 0.  RNK_KGRP is the inverse
    procedure of KGRPSELR, i.e., R = RNK_KGRP(N,KGRPSELR(N,K,R),SORTED).
  *************************************************************************/
  local y,k,r,i,npart,kpart,j,m;

  y = x';
```

```
  if not sorted;
    y = sortc(x',1);
  endif;
  k = rows(y);
  r = 0;
  i = 0;
  do until i==k;
    i = i+1;
    npart = n-i;
    kpart = k-i;
    j = y[i];
    m = i;
    do until m==j;
      npart = npart-1;
      m = m+1;
      r = r + combin(npart,kpart);
    endo;
  endo;

  retp(r);

endp;


outfile = "d:\\pnwr\\mgmt groups.dat";
pathname = "d:\\pnwr";

output file=^outfile reset;

screen on;
output off;
rndseed 047663086;
combomax = 10000;

prec1 = 2;
prec2 = 6;

@ Current management grouping of PNWR compartments @
pnwr_x = { 1  8 13 24 32
```

Appendix C.13. *Compartment permutations.G*. Continued.

```
           9 12 21 25
          10 27 30 33
           6 14 23 29
           4  7 18 22
           3  5 15 20
          11 17 26 34
           2 16 19 28 31 };

@ Compartment center points and adjacency info @
compdat = {1     238575.49613    3670940.60053     4   0   0   0   0   0,
           2     243571.61074    3670438.71926     3   4   6  17   0   0,
           3     242793.37100    3669413.40653     2   4  17   0   0   0,
           4     240605.41302    3670163.81781     1   2   3   0   0   0,
           5     252722.88192    3671343.39818    12  18   0   0   0   0,
           6     244573.26256    3670541.32023     2   7   8  16   0   0,
           7     245612.08738    3672012.54875     6   8   9   0   0   0,
           8     245812.70934    3669841.56405     6   7   9  10  16   0,
           9     247436.21326    3671291.68271     7   8  10  11   0   0,
           10    247984.26868    3669721.95402     8   9  11  13  14   0,
           11    249398.92255    3671601.01051     9  10  12  13   0   0,
           12    251019.09796    3670936.44666     5  11  13  18   0   0,
           13    249733.02444    3669211.44842    10  11  12  14  18  25,
           14    248109.47172    3667570.82865    10  13  15  16  25   0,
           15    247985.92075    3666122.62906    14  16  17  21  22  25,
           16    245508.82922    3667411.36923     6   8  14  15  17   0,
           17    244209.79233    3666622.69556     2   3  15  16  19   0,
           18    252749.65596    3668357.00208     5  12  13  25  27   0,
           19    244877.69627    3664086.85276    17  20  21   0   0   0,
           20    245714.7637     3661802.93       19  21  24   0   0   0,
           21    247056.47912    3663822.86893    15  19  20  22  24   0,
           22    249231.78398    3664660.51488    15  21  23  26   0   0,
           23    250398.05615    3663165.48375    22  24  26  34   0   0,
           24    247919.02743    3662384.20719    20  21  23  34   0   0,
           25    250960.14736    3667292.16841    13  14  15  18  26  27,
           26    251667.61915    3665204.16653    22  23  25  27  28   0,
           27    253663.1848     3666170.544      18  25  26  28   0   0,
           28    254092.41764    3664997.76320    26  27   0   0   0   0,
           29    250118.87738    3658200.61016    30  32  33   0   0   0,
           30    252020.22758    3657656.41924    29  31   0   0   0   0,
           31    252101.17989    3659785.56478    30   0   0   0   0   0,
           32    250353.06475    3654845.33755    29  33   0   0   0   0,
           33    249168.12189    3656121.08301    29  32   0   0   0   0,
           34    248803.71808    3661859.22726    23  24   0   0   0   0};
```

```
d = compdat[.,1 4:9];
xy = compdat[.,1:3];

@ Size of treatment groups @
grpsizes = {1 5,
            2 4,
            3 4,
            4 4,
            5 4,
            6 4,
            7 4,
            8 5};


tymstart = date;
tyminit = tymstart;

allcomp = rows(d);         @ Total number of compartments @
dcols = cols(d);           @ Maximum adjacencies for any compartment @
grpmax = rows(grpsizes);   @ Number of treatment groups @

indices = cumsumc(grpsizes[.,2]);
indices = (indices-grpsizes[.,2]+1) ~ indices;

@ Translate PNWR management group vector into permutation codes @
pnwr_y = {};
i = 0;
do while i<grpmax;
  i = i+1;
  ind = indices[i,.];
  comps = pnwr_x[ind[1]:ind[2]];
  pnwr_y = pnwr_y ~ rnk_kgrp(allcomp,comps,1);
endo;

format /rd 4,0;
goodones = pnwr_x;       @ Matrix of suitable compartment permutations @
goodcode = pnwr_y;       @ Matrix of permutation codes @

k = 0;         @ Counter of suitable compartment lists @
tried = 0;     @ Counter of search trials @

do while k<combomax;
```

Appendix C.13. *Compartment permutations.G*. Continued.

```
tried = tried+1;

@ Draw a candidate permutation of compartments @
x = rndu(allcomp,1);
x = rankindx(x,1)';

@ Search for adjacency conflicts in each treatment grouping @
groups = 0;
ok = 1;
do while groups<grpmax;

  groups = groups+1;

  @ Extract set of compartments for this management group @
  ind = indices[groups,.];
  comps = x[ind[1]:ind[2]];

  @ Create a listing of adjacent compartments for this group @
  adjacent = d[comps,2:dcols];
  adjacent = vecr(adjacent);

  @ MATCH matrix should be = 0 if no adjacent compartments are in group @
  match = (adjacent.==comps);

  if not match==0;
    @ Some compartments in group adjacent to each other -- leave loop @
    ok = 0;
    break;
  endif;

endo;

if ok==1;
  @ This permutation vector passes adjacency tests @

  @ Sort group lists and translate vector into vector of permutation codes @
  y = {};
  i = 0;
  do while i<grpmax;
    i = i+1;
    ind = indices[i,.];
    comps = x[ind[1]:ind[2]];
    comps = sortc(comps',1)';
```

```
  x[ind[1]:ind[2]] = comps;
  y = y ~ rnk_kgrp(allcomp,comps,1);
endo;

@ Has this permutation been previously selected? @
chosen = 1;
i = 0;
do while i<grpmax;
  i = i+1;
  @ Is Y(i) different from every element in i_th column of GOODCODE? @
  if not sumc(goodcode[.,i].==y[i]);
    @ Yes:  Y cannot be equal to any row in GOODCODE -- leave loop @
    chosen = 0;
    break;
  endif;
endo;     @ Y could still be redundant, evaluate next element of Y @

if chosen;
  @ A match was found in every column of GOODCODE, but no proof yet @
  @  that duplicate exists -- check more carefully                  @

  really = 0;
  i = 0;
  do while i<rows(goodcode);
    i = i+1;
    if y==goodcode[i,.];
      @ Perfect match -- duplicate really exists, exit loop @
      really = 1;
      break;
    endif;
  endo;    @ No duplicate so far, read next row of GOODCODE @

  if not really;
    @ Duplicate does not exist, permutation OK to add to GOODCODE @
    chosen = 0;
  endif;

endif;

if not chosen;
  @ This vector has not been previously evaluated @

  k = k+1;     @ Increment number of valid permutations @
```

Appendix C.13. *Compartment permutations.G*. Continued.

```
      @ Update time monitors @
      tymnow = date;
      elapse1 = ethsec(tymstart,tymnow)/100;
      elapse2 = ethsec(tyminit,tymnow)/100;
      tymstart = tymnow;

      if 1 /*k%100==0*/ ;
        @ Print progress @
        format /rdn 4,0;
        print "Trial " tried ", found " k;;
        format /rdn 10,8;
        print ", fraction " (k/tried);;
        format /rdn 8,2;
        print ", Elapsed = " elapse1 " sec, Total elapsed = " elapse2 " sec";
      endif;

      @ Append permutation vector to GOODONES @
      goodones = goodones | x;
      goodcode = goodcode | y;

    endif;

  endif;

endo;

@ Output GOODONES to file @
outform1 = "O*.*lf "~prec1~0;
outform2 = "O*.*lf "~prec2~0;
outformt = ones(rows(allcomp,1)).*outform1 | ones(rows(grpmax,1)).*outform2;
output on;
screen off;
outwidth 160;
call printfm(goodones~goodcode,1,outformt);
output off;
screen on;

@ Are any of the sequences permutations of management groups? @
dups = {};
i = 0;
do while i<rows(goodcode);
  i = i+1;
  vec_i = goodcode[i,.]';
  j = i;
  do while j<rows(goodcode);
    j = j+1;
    vec_j = goodcode[j,.]';
    x = setdif(vec_i,vec_j,1);
    if x==miss(0,0);
      dups = dups | (i~j);
    endif;
  endo;
endo;
print "\nGroup permutations in GOODCODE";
print dups;

save path=^pathname goodones, goodcode;
```

Appendix C.14. *Comp_centers.G*. GAUSS program computes the average inter-compartmental distances within management groups and between management groups separated by one year in time for each permutation. Permutations are ranked according to average values of within-year average distances and successive-year average distances. Highest and lowest-scoring permutations according to these criteria are saved in the GAUSS matrix file BESTCODE.

```
@ Calculates areal dispersion and temporal interspersion statistics on
   compartment combinations @


proc (1) = combin(n,k);
  /******************************************************************************
  Computes combinatorial function for integers N and K, returns integer COMBIN.
  *****************************************************************************/
  local combin,diff,x,i;

  if k==0 or k==n;
    combin = 1;
  else;
    diff = n-k;
    if k<diff;
      x = k;
    else;
      x = diff;
    endif;
    combin = 1;
    i = 0;
    do while i<x;
      i = i+1;
      combin = combin*((n-i+1)/i);
    endo;
  endif;
  combin = round(combin);
  retp(combin);

endp;


pathname = "d:\\pnwr";
load path=^pathname goodones, goodcode;
combomax = 10000;

@ Compartment center points and adjacency info @
compdat = {1     238575.49613    3670940.60053     4  0  0  0  0  0,
           2     243571.61074    3670438.71926     3  4  6 17  0  0,
           3     242793.37100    3669413.40653     2  4 17  0  0  0,
           4     240605.41302    3670163.81781     1  2  3  0  0  0,
           5     252722.88192    3671343.39818    12 18  0  0  0  0,
           6     244573.26256    3670541.32023     2  7  8 16  0  0,
           7     245612.08738    3672012.54875     6  8  9  0  0  0,
           8     245812.70934    3669841.56405     6  7  9 10 16  0,
           9     247436.21326    3671291.68271     7  8 10 11  0  0,
           10    247984.26868    3669721.95402     8  9 11 13 14  0,
           11    249398.92255    3671601.01051     9 10 12 13  0  0,
           12    251019.09796    3670936.44666     5 11 13 18  0  0,
           13    249733.02444    3669211.44842    10 11 12 14 18 25,
           14    248109.47172    3667570.82865    10 13 15 16 25  0,
           15    247985.92075    3666122.62906    14 16 17 21 22 25,
           16    245508.82922    3667411.36923     6  8 14 15 17  0,
           17    244209.79233    3666622.69556     2  3 15 16 19  0,
           18    252749.65596    3668357.00208     5 12 13 25 27  0,
           19    244877.69627    3664086.85276    17 20 21  0  0  0,
           20    245714.7637     3661802.93       19 21 24  0  0  0,
           21    247056.47912    3663822.86893    15 19 20 22 24  0,
           22    249231.78398    3664660.51488    15 21 23 26  0  0,
           23    250398.05615    3663165.48375    22 24 26 34  0  0,
           24    247919.02743    3662384.20719    20 21 23 34  0  0,
           25    250960.14736    3667292.16841    13 14 15 18 26 27,
           26    251667.61915    3665204.16653    22 23 25 27 28  0,
           27    253663.1848     3666170.544      18 25 26 28  0  0,
           28    254092.41764    3664997.76320    26 27  0  0  0  0,
           29    250118.87738    3658200.61016    30 32 33  0  0  0,
           30    252020.22758    3657656.41924    29 31  0  0  0  0,
           31    252101.17989    3659785.56478    30  0  0  0  0  0,
           32    250353.06475    3654845.33755    29 33  0  0  0  0,
           33    249168.12189    3656121.08301    29 32  0  0  0  0,
           34    248803.71808    3661859.22726    23 24  0  0  0  0};

d = compdat[.,1 4:9];
p = compdat[.,1:3];

@ Size of treatment groups @
```

Appendix C.14. *Comp_centers.G*. Continued.

```
grpsizes = {1 5,
            2 4,
            3 4,
            4 4,
            5 4,
            6 4,
            7 4,
            8 5};
sizes = {1 1 1 1 1 2 2 2 2 3 3 3 3 4 4 4 4 5 5 5 5 6 6 6 6 7 7 7 7 8 8 8 8 8};

indices = cumsumc(grpsizes[.,2]);
indices = (indices-grpsizes[.,2]+1) ~ indices;

grpmax = rows(grpsizes);   @ Number of treatment groups @

dist = sqrt((p[.,2]-p[.,2]')^2 + (p[.,3]-p[.,3]')^2);
tdist = abs(sizes'-sizes);
maxes = floor(grpmax/2)*ones(rows(tdist),cols(tdist));
diff = maxes-tdist;
tdist = substute(tdist,diff.<0,maxes+diff);

results = {};
i = 0;
do while i<rows(goodones);
  i = i+1;
  goodrow = goodones[i,.];
  timedist = (tdist.>0);
  timedist = (timedist.>0).*(timedist./tdist);
  timedist = missrv(timedist,0);
  timedist = dist[goodrow,goodrow].*timedist;
  intrsprs = sumc(sumc(timedist.*(tdist.==1)))/sumc(sumc(tdist.==1));
  meandist = 0;
  j = 0;
  do while j<grpmax;
    j = j+1;
    ncomps = grpsizes[j,2];
    ind = indices[j,.];
    group = goodrow[ind[1]:ind[2]];
    grpdist = sumc(sumc(dist[group,group]))/combin(ncomps,2)/2;
    meandist = meandist+grpdist;
  endo;
  meandist = meandist/grpmax;
  results = results | (i~meandist~intrsprs);
endo;
results = results[.,1 2]~rankindx(results[.,2],1)~
          results[.,3]~rankindx(results[.,3],1);
results = results~goodones~goodcode;

bestcode = results[1,40:47];

crit1 = (results[.,3]+results[.,5])/2;
res1 = sortc(crit1~results,1);
format /rdn 7,0;
print "RESULTS matrix sorted by average rank for MEANDIST and INTRSPRS";
print (rankindx(res1[.,1],1)~res1[.,1:48]);

bestcode = bestcode | res1[1:2 combomax:combomax+1,41:48];

crit1 = results[.,3]-results[.,5];
res1 = sortc(crit1~results,1);
format /rdn 7,0;
print "RESULTS matrix sorted by difference in ranks for MEANDIST and INTRSPRS";
print (rankindx(res1[.,1],1)~res1[.,1:48]);

bestcode = bestcode | res1[1:2 combomax:combomax+1,41:48];

save path=^pathname bestcode;

format 7,0;
print bestcode;

results = sortc(results,2);
format 5,0;
print results;

results = sortc(results,4);
print results;

results = sortc(results,6);
print results;
```

APPENDIX D

OVERSTORY DECISION TABLE

Optimal regeneration amounts for 14,641 distributions of forest overstory in the Piedmont National Wildlife Refuge, estimated under three alternative model scenarios, are displayed in a linked table. Conversion between overstory indices and proportions are described earlier (Chapter 4, *Overstory Transition Model*) and are encoded in ASDP programs (Appendix A.9-A.11). Optimal amounts for each type (P2, P3, or P4) are expressed as proportion of the type available Refuge-wide. Model alternatives are those of intermediate (model F0), rapid (model F1), or slow (model F2) hardwood encroachment.

APPENDIX E

MANUSCRIPT: A GENETIC ALGORITHM FOR DYNAMIC OPTIMAL CONTROL OF WILDLIFE HARVESTS

This manuscript describes use of the genetic algorithm to estimate optimal control policies for harvest management of a waterfowl population. I compared the solution obtained by the genetic algorithm to that obtained by an exact procedure, stochastic dynamic programming. I also compared performances by the two algorithms over a simulated management time horizon. The manuscript is in form for submission to the journal *Ecological Modelling*.

A genetic algorithm for dynamic optimal control of wildlife harvests

Clinton T. Moore[a,*], Michael J. Conroy[b], Kevin Boston[c], and Walter D. Potter[d]

[a]*Georgia Cooperative Fish and Wildlife Research Unit, Warnell School of Forest Resources, University of Georgia, Athens, GA 30602, USA*

[b]*U.S. Geological Survey, Biological Resources Division, Georgia Cooperative Fish and Wildlife Research Unit, Warnell School of Forest Resources, University of Georgia, Athens, GA 30602, USA*

[c]*Warnell School of Forest Resources, University of Georgia, Athens, GA 30602, USA*

[d]*Department of Computer Science, University of Georgia, Athens, GA 30602, USA*

*Corresponding author. Tel.: 1-706-5423900; fax: 1-706-5428356.

E-mail address: cmoore@smokey.forestry.uga.edu

Moore et al., Genetic algorithm for wildlife harvests

Abstract

Dynamic programming (DP) methods are often used to find optimal stationary decision policies for dynamic systems, such as the harvesting of wildlife populations. However, DP quickly becomes computationally expensive as the manager contemplates large system models, and DP is altogether infeasible if the modeled decision process is not Markovian. At some point, the manager must be willing to compromise on the pursuit of optimality in order to entertain complexity in a system model. We explored the use of the genetic algorithm (GA) as an alternative to DP. The GA is a stochastic search procedure that easily accommodates large system models, and it has been shown to efficiently locate near-optimal solutions in a variety of problem settings. We selected a problem in waterfowl harvest management to describe the implementation of the GA and to compare state-specific (duck abundance and habitat conditions) decision policies provided by both DP and GA. For two waterfowl models, we found virtually perfect agreement between DP and GA policies. For two others, we found persistent differences in decisions between the two policies at extreme ranges of waterfowl abundance. We used one of these latter two models in a simulation experiment to compare the performance of policies derived by DP, GA, and a random search. Despite differences between the DP and GA decision policies, simulation of the two policies provided almost identical expected values of the objective (cumulative harvest), particularly when the simulated time horizon extended beyond three years. We believe that managers will find the GA to be an attractive alternative to DP when attaining true optima can be compromised with preserving model realism.

Moore et al., Genetic algorithm for wildlife harvests

Keywords: *Anas platyrhynchos*; Dynamic programming; Genetic algorithm; Harvest management; Mallard; Optimal control

1. Introduction

Many natural systems are manipulated for desired sustained outputs, where managers make recurrent decisions in the face of stochastic system behavior. One common example of decision making in natural resource management is that of wildlife harvesting. Harvest decisions are based in part on current population demographics, and decisions generally influence population demographics at the next decision opportunity. Thus the harvest decision determines both the immediate harvest return in the current time period and the nature of all future harvest decisions. Optimal control methods are used to find decision policies for dynamic systems that take these long-term effects of decision making into account.

There are several approaches for deriving optimal control policies for wildlife harvesting. Dynamic programming (DP) is probably the most flexible approach in terms of handling complex systems with some ease (Williams, 1989). Despite its power, DP suffers from explosive growth in computational burden (both time and memory) as problem size increases. Unfortunately, the level of system detail and management control required in some management settings can easily overwhelm DP, forcing one either to concede on problem size or to find alternative approaches.

We explored the use of the genetic algorithm (GA) as an alternative to DP for finding optimal control policies for wildlife harvesting. To test our approach, we chose a problem in waterfowl harvesting to which DP has been successfully applied (Johnson et al., 1997). This system was characterized by two sources of stochasticity: environmental stochasticity and partial controllability, i.e., chance disagreement between the harvest decision and the realized outcome of that decision. We used a simulation experiment to compare performance of a decision policy provided by the GA to an optimal policy provided by DP.

2. Approaches to dynamic optimal control

Williams (1989) provided a comprehensive review of mathematical and computational approaches to optimal control of dynamic systems. He reviewed six classes of methods: variational mathematics, linear programming, nonlinear programming, DP, classical optimization, and simulation. Of these, he identified DP as the approach most capable of accurately accommodating system complexity with a minimum of mathematical difficulty.

2.1. Dynamic programming

DP employs a clever computational approach that significantly simplifies the optimization of a Markovian decision process. Consider a deterministic dynamic system that is in state $\boldsymbol{X}_t$ at time t ($\boldsymbol{X}_t$ may be multidimensional), and suppose that we are faced with selecting a decision d^*_t from the decision set $D_t \in \{d_{t1}, d_{t2}, ..., d_{tK}\}$ that maximizes some return (say, cumulative harvest) from the system defined over the time period t, $t+1$, ..., T. We will assume that the decision process is Markovian, that is, future states of and future returns from the system depend on the past only through the current system state and the decision made in that state (Puterman, 1994). Suppose that we happen to know that the sequence of decisions $D^* = \{d^*_{t+1}, d^*_{t+2}, ..., d^*_T\}$ applied to the system starting at time t+1 when the system is in state $\boldsymbol{X}_{t+1}$ maximizes our objective. Then any decision d_t at t takes us to some future state $\boldsymbol{X}_{t+1}$, for which we know the optimal subsequent decision sequence (by assumption). Bellman (1957) pointed out that the optimal decision, d^*_t, is that decision which maximizes the return over the time interval (t, t+1) plus the return over the interval (t+1, T), given that the system is in state $\boldsymbol{X}_t$.

This insightful "Principle of Optimality" by Bellman (1957) induces a solution algorithm for a dynamic problem. Considering the state of the system in discrete units, one first defines all the possible states of the system and all available decision alternatives at time T-1. Next, the optimal decision $d^*_{T\text{-}1}$ that maximizes the system return (call this maximum value $J_{T\text{-}1}(\boldsymbol{X}_{T\text{-}1})$) over the single time step (T-1, T) is identified for every possible value of $\boldsymbol{X}_{T\text{-}1}$. This is a simple enumeration over all combinations of states and decision alternatives, and the set of optimal decisions thus derived constitutes a state-specific *decision policy* for time T-1. Once the $d^*_{T\text{-}1}$ are identified, the procedure moves backward one time step to the states $\boldsymbol{X}_{T\text{-}2}$. Now the optimal decision $d^*_{T\text{-}2}$ at time T-2 for each state $\boldsymbol{X}_{T\text{-}2}$ is that decision that maximizes the system return over the single time step (T-2, T-1) plus $J_{T\text{-}1}(\boldsymbol{X}_{T\text{-}1})$. Again, this is a simple enumeration over the combinations of states and decisions at T-2. The algorithm may be generalized to accommodate stochastic systems. In this case, a single state-decision combination is evaluated over multiple stochastic outcomes, and an expected system return is calculated.

One may continue stepping backwards through time in this manner for any number of time steps. In particular, the state-specific decision policy at a certain time step (or *decision stage*) t may be compared against the policy previously derived at stage *t*+1. The backwards recursion may be stopped when the policies no longer differ over an arbitrary succession of stages, say, *c* of them. The policy thus derived may be considered to be *stationary*, unchanging with additional decision stages. Other DP computational approaches (bound improvement, policy improvement, linear programming) directly provide optimal stationary policies for Markovian systems (Williams, 1982; Ross 1983; Williams, 1988). Stationary policies are highly desirable in wildlife harvest management, because they imply that the best harvest decision for a given state of the system (i.e., the status of the population and the environment) is constant, regardless of when that state occurs in the overall management time frame.

This backwards-recursion approach prevents one from having to calculate all possible decision pathways forward through time, thus it greatly simplifies the problem. Nevertheless, all DP algorithms are enumerative techniques, thus they are ultimately limited by computational resources. State variables, decision variables, and stochastic inputs are all represented in discrete form in DP. Backwards-recursion DP evaluates every possible combination of these variables at each decision stage. Therefore, increases in problem size, either in the form of more variables or greater degree of discretization of some variables, lead to a geometric increase in computation. Depending on the application, either time or memory may be the limiting computer resource.

The level of realism that natural resource managers often want in their system models may prove infeasible for DP. For example, a few state variables reflecting population status, habitat conditions, and license sales, in combination with other variables representing stochastic environmental influences can severely challenge the DP algorithm: Bellman (1957) referred to this as the “curse of dimensionality.” If any of these variables are changed in scale, for example, if resources are measured and decisions are made in two or more management zones, then the problem may easily overwhelm DP. Furthermore, the DP algorithm is altogether inapplicable if the modeled system is not Markovian. That is, any kind of “memory” property of the system that alters the transition from state $\boldsymbol{X}_t$ to some future state, such as a lag effect of population size on future population growth, renders the problem unsuitable for DP.

2.2. Simulation approaches

If conceding system detail is not a viable option, the decision maker must find a computing alternative. Simulation approaches were also described by Williams (1989). Their main advantage is that they allow one to model a system of substantial complexity and dimension, but their drawback is that their solutions are not necessarily optimal. One popular simulation approach is to randomly search the decision space in a trial-and-error fashion and stop the search after a predetermined number of trials. This search process is “uninformed” in the sense that neither poor nor good guesses are of any influence in determining where to next sample from the decision space. This characteristic makes typical random search procedures highly inefficient for many problems (Goldberg, 1989).

The GA (Goldberg, 1989) is a stochastic approach that iteratively samples the decision space, but unlike uninformed random search, its efforts are concentrated in regions of the decision space that provided good solutions in previous trials. The GA provides no guarantee of optimality, but its searching algorithm provides assurance that its solution is nonetheless satisfactory for the finite level of resources one has to expend toward the solution.

3. A waterfowl harvesting problem

3.1. System description

The mallard (*Anas platyrhynchos*) is an abundant waterfowl species that is widely distributed throughout North America and much of the rest of the world (Bellrose, 1976). On average, 7.4 million ducks (range 5.0-11.2 million; 1955-1999 data; U. S. Fish and Wildlife Service, 1999b) breed in a 3.6 million-ha region of the mid-continent. The breeding habitat includes estuaries, tundra, and boreal forests in Alaska and northwestern Canada; boreal forests in the western half of Ontario and the northern portions of Alberta, Saskatchewan, Manitoba; prairie potholes in the southern portions of these provinces; and grasslands in the northcentral U. S. (Anderson and Henny, 1972). These habitats are surveyed by air and by ground annually in May to estimate numbers of breeding ducks (Smith, 1995).

Annual mallard productivity is partially density-dependent but is also strongly tied to abundance of wetland breeding areas in the prairie pothole region (Pospahala et al., 1974). Ponds in southern Canada are also counted during the waterfowl surveys, and their abundance averaged 3.5 million (range 1.4-6.4 million) between 1961-1999 (U. S. Fish and Wildlife Service, 1999b). Pond abundance is influenced by annual precipitation in the region (Pospahala et al., 1974), which averaged 418 mm recorded at 5 weather stations in southern Canada in the period 1942-1991 (range 304-574 mm; Johnson et al., 1997).

The mallard is subject to sport hunting in North America and usually comprises the largest proportion of all waterfowl harvested annually (Anderson and Henny, 1972; Martin and Carney, 1977). Consequently, the setting of annual harvest regulations for this species generates considerable interest among biologists, wildlife agency administrators, hunters, and the general public. In the United States, the U.S. Fish and Wildlife Service, in cooperation with state wildlife management agencies and other participants, sets annual harvest regulations for mallards and other migratory bird species (Martin et al., 1978). Though several studies have addressed the optimal control of mallard harvests (Anderson, 1975; Williams, 1988; Williams, 1996b; Johnson et al., 1997), formal optimization methods have only recently (1995) been introduced to the regulations setting process. We will briefly review that model, which was presented by Johnson et al. (1997).

3.2. Population dynamics model

Johnson et al. (1997) described mallard population size in year $t+1$, N_{t+1}, as a nonlinear function of N_t, a habitat measurement P_t (number of ponds in prairie Canada at time t), and a harvest rate decision h_t:

$$N_{t+1} = N_t + f_i(N_t, P_t, h_t).$$

The function f is subscripted to denote structural uncertainty about the behavior of the system, i.e., that there exist alternative, plausible models that describe the population dynamics of mallards; this will be elaborated later. The habitat state variable P_{t+1} was itself assumed to be a linear function of P_t and a stochastic rainfall total, r_{t+1}:

$$P_{t+1} = s(P_t, r_{t+1}).$$

Four harvest decision alternatives were available: either (1) season closure, or bag limit and season length combinations considered relatively (2) restrictive, (3) moderate, or (4) liberal. These alternatives corresponded to mean harvest rates h_t of 0, 0.090, 0.120, and 0.156, respectively. However, the harvest rate outcome of an enacted regulatory decision was random, occurring with mean h_t. Thus, the harvest process was only partially controlled.

The objective that Johnson et al. (1997) maximized for each alternative model i was

$$J_i = \sum_{\tau=t}^{T} u(N_{i,\tau+1})H_{i,\tau}$$

where $H_{i,\tau}$ is a model-specific harvest at time τ and $u(N_{i,\tau+1})$ is a piecewise linear utility function of next year's expected population size. The utility function is valued between 0 and 1 and weights harvest according to expected population size. These models and this objective function define a Markovian decision process, and they imply that the search for optimal harvest rate decisions must take into account all future harvests and future environmental conditions.

Johnson et al. (1997) addressed the issue of structural uncertainty by considering four alternative models of population response to harvest and the environment. Perhaps the greatest source of uncertainty concerns the response of the mallard population to harvest. The hypothesis of compensatory harvest mortality (S_C) states that below some threshold rate of harvest, increases in other sources of mortality compensate for decreases in harvest mortality (Nichols et al., 1984). Stated another way, harvest mortality may increase up to some threshold point with no corresponding increase in total mortality.

Alternatively, the additive harvest mortality (S_A) hypothesis states that no such compensation occurs: each incremental increase in harvest mortality “adds” to the background mortality already occurring (Nichols et al., 1984). Both hypotheses have been supported in retrospective studies of mallard data (Anderson and Burnham, 1976; Smith and Reynolds, 1992). Unfortunately, experiments that can distinguish the veracity of either hypothesis are lacking because they are not easily designed or executed (Nichols et al., 1984; Nichols and Johnson, 1989).

A second important source of uncertainty concerns mallard production in the spring. Production is believed to be directly correlated to habitat conditions but inversely related to population abundance (Pospahala et al., 1974). Johnson et al. (1997) considered two cases regarding the strength of this latter relationship: production of young was either weakly (R_W) or strongly (R_S) depressed by increasing population abundance.

Combinations of contrasting hypotheses about additivity (S_C, S_A) and density dependence (R_W, R_S) yielded four alternative models of population response: S_CR_W, S_CR_S, S_AR_W, S_AR_S. Each of the models was stochastic, incorporating random processes for pond dynamics and harvest outcomes. Johnson et al. (1997) used program SDP (Lubow, 1995), a backwards-recursion algorithm, to find optimal stationary decision policies for each of the models separately and for a composite model that weighted each individual model by a probability or belief weight. Each decision policy provided a model-specific optimal regulatory (harvest rate) decision for each discrete combination of N_t and P_t.

4. The genetic algorithm

The GA seeks to solve combinatorial optimization problems through an analogy to evolutionary biology. By simulating natural selection in a collection of artificial organisms, the procedure searches for the “most fit” organism in the set. The chromosome of each organism carries a genetic code, or genotype, that represents 1 candidate solution to the decision problem. As with biological chromosomes, the chromosomes of different organisms may be recombined in offspring. Furthermore, any bits of genetic code, or alleles, in the offspring may be mutated at random. Recombination and mutation then result in the appearance of novel genotypes in the population at each generation. Various rules govern the stochastic processes of pairing, recombination, and mutation of genetic material. In general, the “fitness” of an organism (the objective value corresponding to the encoded candidate solution) determines the probability that its genetic material persists, either in complete or in partial form, in the next generation. The particular strength of the GA is that these stochastic operators sometimes produce offspring with greater fitness than either of the parents. By the end of this natural selection process, an organism emerges with substantially greater fitness than the individuals in the initial generation.

The GA is a random search procedure, but its probabilistic forays into the decision space are guided by real-time search results. The GA is described as an efficient, robust procedure that provides solutions achieving a good compromise between the identification of bona fide optimality and the accommodation of large problems (Goldberg, 1989). In control problem applications, the GA offers distinct advantages over DP. First, the GA conducts forward simulations over the entire problem time frame, which, for most practitioners, may be easier to comprehend than the backwards-progressing single-step calculations of DP. Second, because the GA forward-simulates, the decision process need not be Markovian. Third, there is no need to discretize the problem as is done with DP because the GA performs no enumerations. For this reason, computational burden for the GA grows linearly with problem size (Michalewicz et al., 1992) rather than geometrically, as is the case for DP. However, enumeration allows DP to provide optimal decisions for a range of initial system states in a single run; in contrast, a single run of the GA provides a decision only for a single system state.

5. A genetic algorithm for waterfowl harvest

5.1. Methods

As described above, the DP algorithm solves a dynamic control problem by stepping backwards through time, moving away from the end of the decision time frame. Eventually, the "end-of-time" effect has a diminishing influence on the identification of the optimal decision. When successive time-dependent decision policies no longer differ from each other, the algorithm considers the current policy as "stationary" and thus stops.

Our approach concerned finding an optimal schedule of decisions starting from a given initial system state. By virtue of the fact that long time horizons attenuate the "end-of-time" effect, we reasoned that if it was possible to find a superior schedule for a sufficiently long time frame, say T years, then the first (time-1) decision value in that schedule should be comparable or identical to the value obtained from the stationary optimal policy.

Therefore, we used the GA to search for an optimal harvest decision schedule over a fixed time frame for an initial system state and system model. We chose $T = 15$ years subjectively but after observing that time-1 decision values were consistent beyond $T \approx 7$ years in other GA runs. We applied the GA to the four stochastic models of Johnson et al. (1997). However, our decision set comprised 81 levels of harvest rate rather than four: $h_t \in \{0.0, 0.00625, 0.0125, ..., 0.50\}$. Additionally, we sought to maximize a long-term sum of unweighted harvests instead of utility-weighted harvests.

We represented each annual decision opportunity as one gene on a 15-gene chromosome. Each gene took on one of 81 possible integer-valued alleles, representing harvest decision values. Thus, a chromosome carrying a specific complement of 15 alleles (the "genotype") encoded a particular 15-year harvest decision schedule, and the first allele A_1 on the chromosome represented the decision value at time 1 of the schedule.

Because each chromosome encoded a particular harvest schedule, a value of total harvest was obtained when the schedule and an initial system state were input to one of the alternative mallard models. However, because the mallard models were stochastic, total harvest values were not unique for a given schedule. Therefore, we replicated each schedule m times and obtained an expected value of total harvest. This expectation served as the fitness value for the chromosome (Fitzpatrick and Grefenstette, 1988). The stochastic nature of the chromosome evaluation implied that chromosomes with identical genotypes may receive different fitness values.

In the simplest implementation of a GA (Goldberg, 1989), a fixed population of C artificial organisms is followed over a fixed number of generations, G. The organisms pair up, exchange genetic material, then are replaced by two offspring. To make pairing assignments, we first replaced fitness values F for each organism by a linear function of population rank (Davis, 1991, pp. 31-34):

$$F' = \max(k_1, k_2 - k_3(C - \mathrm{rank}(F))),$$

where k_1 is a minimum value for the function, k_2 is a maximum value, and k_3 is the step size of the function. Rank-based fitness evaluation prevents a "super" organism from quickly dominating the population (i.e., rapid convergence to a local optimum) (Davis, 1991). We summed the F' values over the population, and we created a cumulative distribution of the proportionate contribution of each F' to the sum. Based on this distribution, we randomly drew two individuals at a time to serve as a parental pair ("roulette wheel" selection; Goldberg, 1989). All selection was done with replacement, thus not only could a single individual appear in more than one parental pair, it could also be paired with itself. Each pair was placed in a mating pool of size $C - E$, $0 \leq E < C$, where E is a fixed number of individuals from the parent population chosen to persist into the next generation (described below). Pairs of individuals were selected until the mating pool was filled.

After pair selection, pair members were chosen for exchange of genetic material. For each pair, the outcome of a Bernoulli trial with probability P_C determined whether "crossover" of genetic material occurred. Following a successful trial, two loci (allele sites), L_1 and L_2, $1 \leq L_1 < L_2 < 15$, were chosen at random. The set of alleles between the first locus and the L_1 locus and between the (L_2+1) locus and the 15th locus in parent 1 were given to child A, and the alleles between the (L_1+1) locus and the L_2 locus in parent

2 were given to child B. The remaining alleles in parent 1 were given to child B and the remainder in parent 2 were given to child A. If the trial was not successful, child A was “cloned” from (inherited the entire genotype of) parent 1, and child B was cloned from parent 2. When an organism was paired with itself, two clones were produced regardless of the Bernoulli trial outcome.

Pairing and crossover produced $C - E$ offspring. For every gene in the offspring population, we conducted a Bernoulli trial with probability P_M to determine mutation outcome. If the trial was unsuccessful, the allele value on the gene was left intact. If the trial was successful, the allele value was “mutated”, or replaced by another one chosen at random.

After the mutation trials, the $C - E$ offspring were combined with the E highest-fit individuals in the parent population to form a replacement population of C members. Such “elite” selection of most-fit parents assured that the best genotypes found so far are retained in the population to participate in further rounds of genetic recombination (Davis, 1991, p. 34). The replacement population then became the new parent population in the next generation.

We implemented a GA for mallard harvesting in the C programming language. The GA performed the following steps:

```
1. Receive as input an initial system state $\boldsymbol{X}_1 = (N_1, P_1)$ and system model
2. Initialize a population of $C$ organisms with $15C$ randomly chosen alleles
3. $g \leftarrow 0$
4. Do until $g = G$
```

```
        1.  Evaluate expected fitness of all organisms
        2.  Construct mating pool
        3.  Crossover genetic material in paired parents
        4.  Mutate alleles of offspring
        5.  Create replacement population from offspring plus elite-selected parents
        6.  $g \leftarrow g + 1$
    11. Retrieve organism with greatest fitness; interpret allele value $A_1$
```

The harvest rate h_1 corresponding to A_1 was an estimate of optimal stationary harvest rate for the initial state $\boldsymbol{X}_1$ and the given model. Because the GA and the population models were stochastic, we repeated the steps above n times for $\boldsymbol{X}_1$. We calculated the median of the h_1 values, and we took this value as our estimate of the optimal stationary harvest rate for the initial state $\boldsymbol{X}_1$. We calculated a 95% confidence interval for the median based on a large-sample approximation (Hollander and Wolfe, 1973, p. 49).

5.2. Policy derivation

We used the GA to estimate optimal harvest decisions for a few initial states under each of the four population models. Our analyses used $C = 100$ individuals per generation followed over $G = 50$ generations. We evaluated fitness for each individual based on $m = 30$ replications of the harvest schedule. Fitness scaling parameters k_1, k_2, and k_3 were fixed at 100, 2, and 2, respectively, and crossover and mutation probabilities were $P_C = 0.8$ and $P_M = 0.1$, respectively. At each generation, the $E = 3$ most fit organisms were directly placed into the next population. Each GA run was repeated $n = 100$ times with a different random seed.

We conducted GA analyses for 10 combinations of initial mallard abundance (4, 6, 8, 10, and 12 million mallards) and initial pond abundance (2 and 6 million ponds). We compared these solutions against optimal decision values provided by program SDP (Lubow, 1995).

5.3. Comparison of GA and DP decision policies

Consistency between the DP and GA solutions varied by model and by initial system state (Fig. 1). Solutions were mostly consistent for the two models of compensatory harvest mortality (S_CR_S and S_CR_W) for all initial system states. Most confidence intervals for the GA solution were relatively small and contained the optimal harvest rate. Only in the higher-abundance initial duck states under model S_CR_S was there a suggestion that GA solutions consistently underestimated optimal harvest rates.

In contrast, solutions were often inconsistent under the two models of additive harvest mortality (S_AR_S and S_AR_W), and the direction of differences depended on initial system state (Fig. 1). For initial states of high duck abundance, the GA solution consistently underestimated optimal harvest rate. For states of very low duck abundance, the GA often overestimated optimal harvest rate.

6. Simulation of alternative decision policies

6.1. Methods

We were concerned by the inability of the GA to find optimal harvest rates for some initial states under the models of additive harvest mortality. The differences persisted despite our tuning of the most fundamental parameters of the GA (T, C, G, m, P_C, P_M).

Despite these differences, we conducted a simulation experiment to compare the performance of a GA-derived decision policy for the S_AR_S model against that of an optimal policy computed by program SDP. We constructed a state-specific decision policy matrix by running the GA $n = 100$ times for each combination of 21 initial population states (2–12 million mallards, by 0.5 million) and 13 initial habitat states (1–7 million ponds, by 0.5 million). Each GA run was based on $C = 200$ individuals and $G = 100$ generations (i.e., 20,000 candidate 15-year decision schedules per run). We used $k_2 = k_3 = 1$ and $E = 5$; all other GA parameters retained their values as before.

We wished to compare performance by the GA and optimal policies against a "baseline" policy derived by uninformed random search (RS). We constructed this policy for the same 21 × 13 array of initial states. For each state, we drew 20,000 candidate 15-year decision schedules completely at random. As in the GA, we calculated an expected value of accumulated harvest based on $m = 30$ replications of each schedule. The first decision value for the schedule providing the greatest expected return was retained, and we performed the search $n = 100$ times per initial state. The median of the 100 harvest rates served as our RS estimate of optimal harvest rate for the given state. The RS and GA approaches performed the same number of harvest schedule evaluations and thus were computationally comparable. In fact, the RS approach is practically identical to the GA described above with $P_M = 1.0$ and $E = 1$.

With 3 comparable state-specific decision policies in hand (optimal, GA, RS), we simulated harvest management under each policy over five time frames: 1, 3, 10, 30, and 100-year. For each time frame T and an initial mallard and habitat state, we conducted

10,000 trials of the S_AR_S model. In each trial, we selected a random series (of length T) of annual rainfall amounts and harvest rate errors. We kept track of mallard abundance and cumulative harvest according to the optimal decision policy (N_t^* and J_t^*, respectively), the GA policy (N_t^{GA} and J_t^{GA}), and the RS policy (N_t^{RS} and J_t^{RS}). We applied decisions from each of the policies at each decision opportunity during the trial. For each policy $l \in \{*, \mathrm{GA}, \mathrm{RS}\}$, current pond abundance and N_t^l indexed a decision value to be applied to N_t^l. The harvest obtained from that decision (following application of the same harvest rate error to all three harvest rates) was accumulated in J_t^l, and the population state was advanced to $N_{t+1}{}^l$. We calculated values J_T^l as the median of the J_T^l over the 10,000 trials. Similarly, we obtained median terminal population sizes $N_{T+1}{}^l$. The quantities $R_T^{(1)} = J_T^{\mathrm{GA}} / J_T^*$ measured the relative performance of the GA policy against that of the optimal policy, and $R_T^{(2)} = (J_T^{\mathrm{GA}} - J_T^{\mathrm{RS}}) / (J_T^* - J_T^{\mathrm{RS}})$ measured the improvement offered by the GA policy relative to the RS policy. We conducted each set of trials over the five time frames and over 20 combinations of initial mallard population size (4, 6, 8, 10, 12 million) and pond abundance (2, 4, 6, 8 million).

6.2. Results

Decisions in the GA-derived policy differed from those in the optimal policy in a pattern similar to that described earlier. Relative to the optimal decisions, GA-derived decisions were consistently greater at low mallard population sizes and consistently smaller at high population sizes (Fig. 2a). Differences from the optimal policy for the RS-derived policy were similar in pattern but more pronounced and more variable than the GA differences (Fig. 2b).

Beyond the briefest time horizon ($T \geq 3$), $R_T^{(1)} > 0.98$ for any initial state; for $T \geq 10$, $R_T^{(1)} > 0.995$ (Fig. 3). In spite of the harvest rate differences between policies, simulated performances by the two policies were practically identical.

The RS policy also performed well. Its average cumulative harvests at 3 years and 10 years were >97% and >99.3% of J_T^*, respectively. In nearly all cases, however, the GA policy outperformed the RS policy (i.e., $R_T^{(2)} > 0$), and the performance disparity increased with length of the time frame (Fig. 4). Both the GA and RS policies were derived under considerable computational effort (2×10^6 candidate decision schedules evaluated per initial state), so the good performance offered by the RS policy could be explained by a sufficiently thorough sampling of the decision space. If the sampling rate were decreased (as might be the case for a larger decision problem), chance encounters of good decisions become less likely. Therefore, we would anticipate that the performance difference between the GA and RS policies would increase in this circumstance, because the GA uses information from the search to increase its chances of finding good solutions. That is, we expect the utility of the GA relative to RS to become more apparent as computational resources become limiting.

For most of the time horizons ($T \geq 3$), the median terminal population size under the GA policy was slightly greater than that under the optimal policy ($N_{T+1}^{\mathrm{GA}} / N_{T+1}^*$ range = 1.006-1.018). Therefore, in terms of the waterfowl resource, the GA policy tended to be risk-aversive relative to the optimal policy for all but the shortest time horizons. For $T = 1$, N_2^{GA} was sometimes smaller than N_2^*, but differences were not substantial ($N_2^{\mathrm{GA}} / N_2^*$ range = 0.978-1.114).

7. Discussion

Our purpose was to illustrate the application of the GA to a type of problem that through its size often defies analysis by conventional optimization algorithms. For this illustration, the models were sufficiently small to enable comparison of this procedure to one known to provide an optimal solution. In practice however, system models and precision requirements may be so demanding that no optimal solution is available.

For example, suppose that one desires the optimal harvest decision given that the current abundances of breeding mallards and ponds are 7 million and 4 million, respectively, and that one of the four harvest models can be considered an appropriate model. The user also specifies ranges for the mallard (2-10 million) and the pond (1-7 million) state variables. If ranges for both state variables are divided into increments of 1 million, then derivation of the optimal policy by a backwards-recursion DP procedure (B. C. Lubow, Colo. State Univ., unpubl. software) is trivial, requiring only 17 seconds on a 200-Mhz desktop computer and using 1.90 Mb in temporary data storage. However, the quality of the DP solution is suspect because the state grid is very coarse and interpolations used by the program could be quite crude. Therefore, the user instead discretizes the state variables into increments of 0.1 million. The solution by DP is still rather accessible, requiring 3934 seconds of CPU time and 767 Mb in storage space. However, resource use has increased by factors of 231 (CPU time) and 404 (storage), whereas problem size increased only by a factor of 78.4.

Thus, increases in problem size brought about by finer discretization of state variables generally requires proportionately larger increases in DP computational resources. In contrast, discretization of state variables and stochastic inputs is not required in a GA, so the GA does not suffer the same explosive growth in computational resource use as does DP. A GA using the parameter values provided earlier (section 5.2) solved the above problem in 7125 seconds. Though the DP was faster for this problem, we point out that this advantage will disappear as the problem is further discretized for DP. To illustrate, values for the stochastic rainfall and harvest rate variables were drawn from discretized versions of continuous distributions, each divided into five probability bins. By increasing the level of discretization from five to 10 bins, for example, we would expect to see a substantial increase in computer resources used by DP, but no increase at all for the GA. In fact, we could draw values directly from the parent distributions themselves under the GA and detect no significant increase in use of computer resources.

Of even greater concern than discretization issues is the desire to extend these models to accommodate greater specificity in mallard stocks and greater regional control of the harvest (U. S. Fish and Wildlife Service, 1999a) or to incorporate habitat management objectives (Johnson et al., 1996). These extensions will almost certainly foil attempts at optimization by classical techniques. In such cases, it is of course impossible to assess the quality of a GA solution because the “yardstick” of optimality is lacking. Therefore, progress on these larger problems will require some assurance that the favorable performance by the GA extends beyond small problems. Through this demonstration, we hope, natural resource managers will develop a sense of comfort with the GA and begin to recognize its utility for the optimal control of stochastic dynamic systems.

One particular extension of the harvesting problem is that of adaptive optimization under model uncertainty (Williams, 1996a). In the example above, we demonstrated the GA for each of four alternative population models. As earlier discussed, harvest managers are not certain about which, if any, of these models are appropriate for management. Under adaptive optimization, relative measures of current belief in each model are expressed in a set of additional state variables. Comparing the result of a decision against that predicted by each model provides a means to update model belief measures through time, where greater credibility is supplied to better performing models and removed from others. This updating mechanism becomes part of the system dynamics in adaptive optimization. Thus, optimal control of such a system not only takes into account how decisions affect future harvests, but also how decisions affect future belief in each model. Consequently, optimal decisions are those that seek to balance short-term system gains against gains of information used to better manage the system over the long run. Adaptive optimization is possible through DP approaches (Williams, 1996b; B. C. Lubow, Colo. State Univ., unpubl. software), but this expansion of the problem poses considerable challenges to enumeration techniques. The GA might be a feasible approximate alternative for adaptive optimization.

Although reproduction, crossover, and mutation are common elements in any GA, there is no “standard” way that these operations are carried out. The operators that we chose are probably among the most common, but many variations exist. Choice of technique is up to the judgment of the practitioner and may be dictated by the problem itself. Furthermore, several parameters often control the operators, and the practitioner must select parameter values. For our illustration, we chose parameter values through informal trial-and-error; successful application of the GA to a control problem may require more formal experimentation of alternative GA parameter settings than the kind we performed.

An alternative mutation operator that recognizes the ordinal nature of alleles might improve convergence of the GA toward superior decisions. Our mutation operator selected any allele from the entire allele set without regard to how long the population had evolved. For example, consider a decision string {0.10, 0.10, 0.12, ...} that has persisted toward the end of a GA run by having performed well through the majority of the generations. Suppose that its first allele is now targeted for mutation. It seems intuitively clear that at this point in the GA, a mutation “nudge” would be far more beneficial to further progress of the GA than would a “kick”. However, our mutation operator would more likely “kick” the solution out of contention with an allele replacement value of, say, 0.45 than “nudge” the solution into possibly greater fit with a replacement value of 0.09 or 0.11. Thus, in the latter generations of the GA, it might make better sense to constrain the allele set to those in the neighborhood of the target allele. Michalewicz et al. (1992) advocated non-uniform probabilistic selection of replacement alleles, and they provided a routine for dynamic adjustment of these probabilities.

8. Conclusion

We have shown that the GA derives reasonable, approximate solutions to one kind of stochastic optimal control problem in natural resources management. The advantages of the GA are that one can maintain a forward-time perspective in solving the problem (thus one is not restricted to assuming Markovian state transitions) and that no enumeration is done (thus state variables and stochastic inputs need not be discretized). Consequently, the GA can accommodate models of substantial complexity relative to DP. Despite the simplicity of our GA implementation and the subjective nature of our GA parameter selection, solutions provided by the GA were consistent with optimal solutions for two of the models we tested. Solutions were suboptimal in two other cases; nonetheless simulation of one of these decision policies provided returns that were not considerably different than those provided by the optimal policy.

The trend in decision making in natural resource management is toward system models having greater realism and toward the acknowledgment and confrontation of system uncertainty. At some point, the cost of these pursuits is the inability to derive bona fide optimal decision policies. The GA and other heuristic procedures including simulated annealing (Kirkpatrick et al., 1983), tabu search (Glover, 1986), and artificial neural networks (Narendra, 1996) may provide reasonable alternatives to classical optimization procedures. For example, in a spatially-explicit silvicultural scheduling problem where the management objective was the conservation of a bird species, Moore et al. (2000) used a GA when the problem could not be satisfactorily simplified for presentation to DP. We believe that decision makers will often be willing to exchange an optimal solution to a simplistic problem for a satisfying, near-optimal solution to a more realistic problem.

Acknowledgments

We appreciate the helpful reviews of this paper provided by W. L. Kendall and J. T. Peterson. The work of CTM was supported by the U.S. Environmental Protection Agency, STAR Graduate Fellowship Program, and by the U.S. Geological Survey, Biological Resources Division (Work Order #34, Coop. Agreement #14-16-0009-1551). The Georgia Cooperative Fish and Wildlife Research Unit is jointly sponsored by the Biological Resources Division, the Georgia Department of Natural Resources, the University of Georgia, and the Wildlife Management Institute.

References

Anderson, D. R., 1975. Optimal exploitation strategies for an animal population in a Markovian environment: a theory and an example. Ecology, 56: 1281-1297.

Anderson, D. R. and K. P. Burnham, 1976. Population ecology of the mallard. VI. The effect of exploitation on survival. U. S. Fish Wildl. Serv. Resour. Publ. 128, 66 pp.

Anderson, D. R. and C. J. Henny, 1972. Population ecology of the mallard. I. A review of previous studies and the distribution and migration from breeding areas. U. S. Fish Wildl. Serv. Resour. Publ. 105, 166 pp.

Bellman, R., 1957. Dynamic programming. Princeton Univ. Press, Princeton, N. J., 342 pp.

Bellrose, F. C., 1976. Ducks, geese, and swans of North America. Stackpole, Harrisburg, Pa., 540 pp.

Davis, L. (Editor), 1991. Handbook of genetic algorithms. Van Nostrand Reinhold, New York, 385 pp.

Fitzpatrick, J. M. and J. J. Grefenstette, 1988. Genetic algorithms in noisy environments. Mach. Learn., 3: 101-120.

Glover, F., 1986. Future paths for integer programming and links to artificial intelligence. Comput. Oper. Res., 13: 533-549.

Goldberg, D. E., 1989. Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, Reading, Mass., 412 pp.

Hollander, M. and D. A. Wolfe, 1973. Nonparametric statistical methods. Wiley, New York, 503 pp.

Johnson, F. A., C. T. Moore, W. L. Kendall, J. A. Dubovsky, D. F. Caithamer, J. R. Kelley, Jr. and B. K. Williams, 1997. Uncertainty and the management of mallard harvests. J. Wildl. Manage., 61: 202-216.

Johnson, F. A., B. K. Williams and P. R. Schmidt, 1996. Adaptive decision-making in waterfowl harvest and habitat management. Proc. Int. Waterfowl Symp., 7: 26-33.

Kirkpatrick, S., C. D. Gelatt and M. P. Vecchi, 1983. Optimization by simulated annealing. Science, 220: 671-680.

Lubow, B. C., 1995. SDP: generalized software for solving stochastic dynamic optimization problems. Wildl. Soc. Bull., 23: 738-742.

Martin, E. M. and S. M. Carney, 1977. Population ecology of the mallard. IV. A review of duck hunting regulations, activity, and success, with special reference to the mallard. U. S. Fish Wildl. Serv. Resour. Publ. 130, 137 pp.

Martin, F. W., R. S. Pospahala and J. D. Nichols, 1978. Assessment and population management of North American migratory birds. In: J. Cairns, Jr., G. P. Patil and W. E. Waters (Editors), Environmental monitoring, assessment, prediction, and management–certain case studies and related quantitative issues. International Co-operative Publishing House, Fairland, Md., pp. 187-239.

Michalewicz, Z., C. Z. Janikow and J. B. Krawczyk, 1992. A modified genetic algorithm for optimal control problems. Comput. Math. Appl., 23(12): 83-94.

Moore, C. T., M. J. Conroy and K. Boston, 2000. Forest management decisions for wildlife objectives: system resolution and optimality. Comput. Electron. Agric., In press.

Narendra, K. S., 1996. Neural networks for control: theory and practice. Proc. IEEE, 84: 1385-1406.

Nichols, J. D., M. J. Conroy, D. R. Anderson and K. P. Burnham, 1984. Compensatory mortality in waterfowl populations: a review of the evidence and implications for research and management. Trans. North Am. Wildl. Nat. Resour. Conf., 49: 535-554.

Nichols, J. D. and F. A. Johnson, 1989. Evaluation and experimentation with duck management strategies. Trans. North Am. Wildl. Nat. Resour. Conf., 54: 566-593.

Pospahala, R. S., D. R. Anderson and C. J. Henny, 1974. Population ecology of the mallard. II. Breeding habitat conditions, size of the breeding populations, and production indices. U. S. Fish Wildl. Serv. Resour. Publ. 115, 73 pp.

Puterman, M. L., 1994. Markov decision processes. Wiley, New York, 649 pp.

Ross, S., 1983. Introduction to stochastic dynamic programming. Academic Press, New York, 164 pp.

Smith, G. W., 1995. A critical review of the aerial and ground surveys of breeding waterfowl in North America. Natl. Biol. Serv. Biol. Sci. Rep. 5, 252 pp.

Smith, G. W. and R. E. Reynolds, 1992. Hunting and mallard survival, 1979-88. J. Wildl. Manage., 56: 306-316.

U. S. Fish and Wildlife Service, 1999a. Adaptive harvest management 1999 duck hunting season. Unpublished report. [http://www.fws.gov/r9mbmo/reports/ahm99/ahm1999.pdf]

U. S. Fish and Wildlife Service, 1999b. Waterfowl population status, 1999. Unpublished report. [http://www.fws.gov/r9mbmo/reports/status99/status99.pdf]

Williams, B. K., 1982. Optimal stochastic control in natural resource management: framework and examples. Ecol. Modell., 16: 275-297.

Williams, B. K., 1988. MARKOV: a methodology for the solution of infinite time horizon Markov decision processes. Appl. Stochastic Models Data Anal., 4: 253-271.

Williams, B. K., 1989. Review of dynamic optimization methods in renewable resource management. Nat. Resour. Model., 3: 137-216.

Williams, B. K., 1996a. Adaptive optimization and the harvest of biological populations. Math. Biosci., 136: 1-20.

Williams, B. K., 1996b. Adaptive optimization of renewable natural resources: solution algorithms and a computer program. Ecol. Modell., 93: 101-111.

Moore et al., Genetic algorithm for wildlife harvests

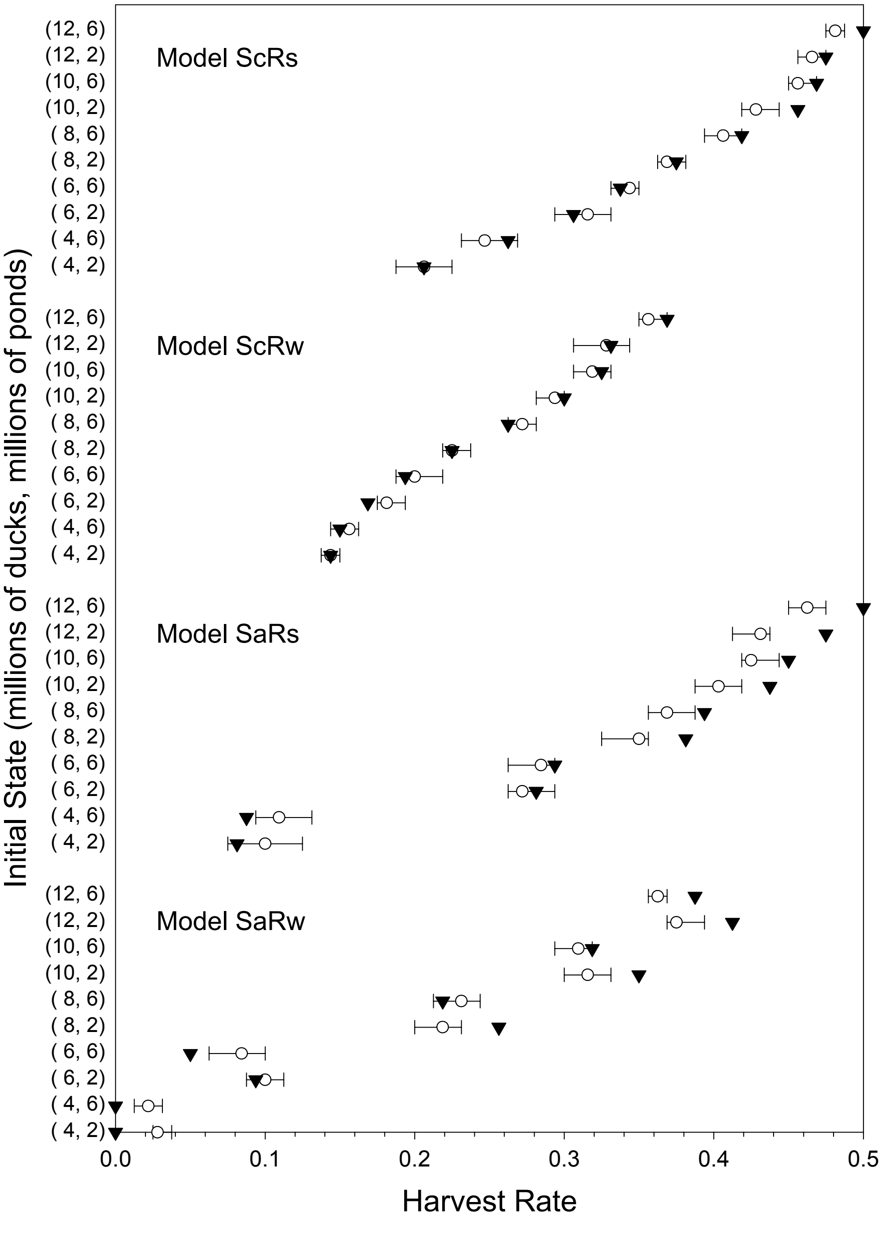
Figure Legends

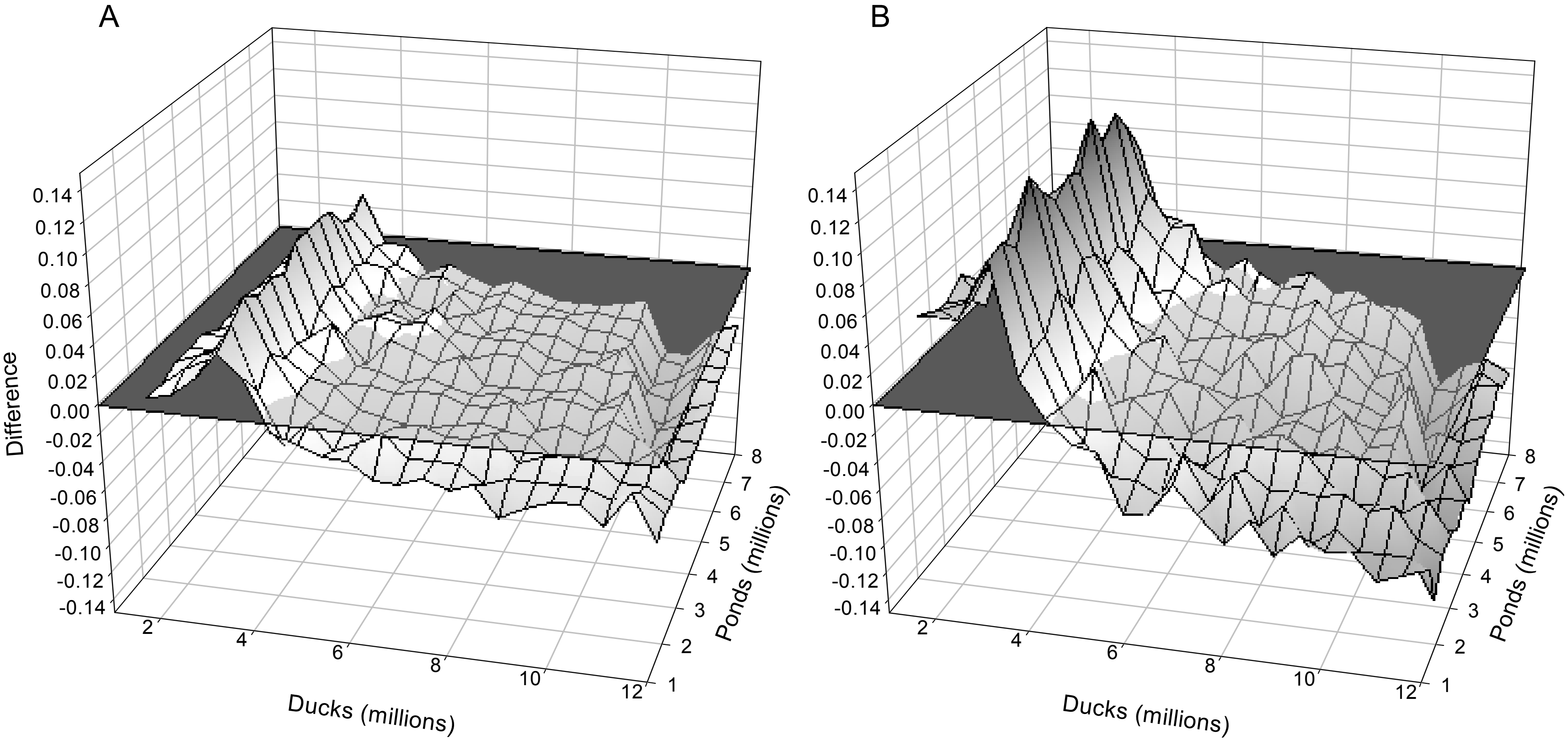
Fig. 1. Estimates (open circles, displayed with 95% confidence intervals) by the genetic algorithm of optimal harvest rates (filled triangles) for 4 stochastic models of mallard population dynamics and 10 initial states of duck and pond abundance (millions).

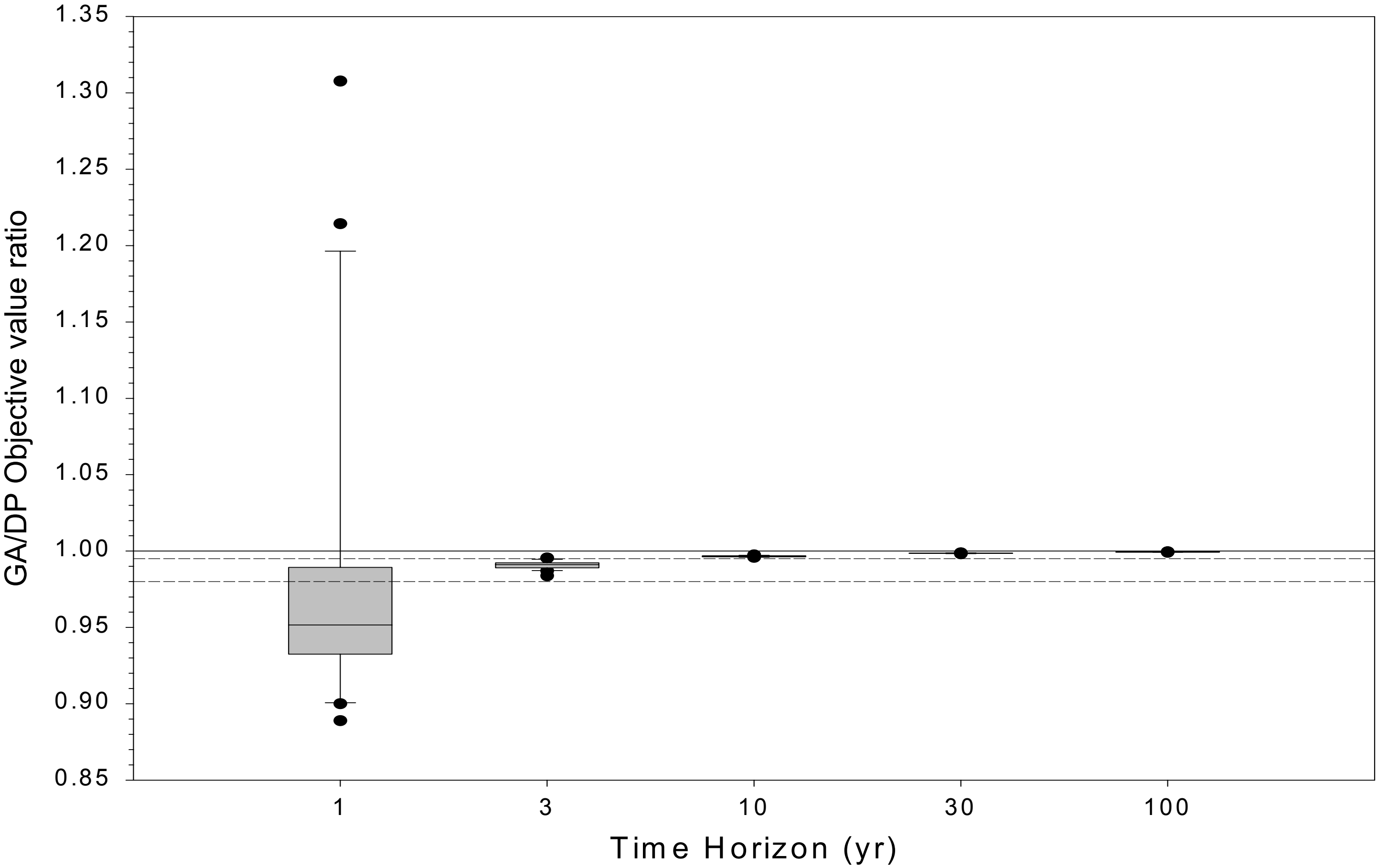
Fig. 2. Optimal harvest rate estimation error (h_{est} - h_{opt}) for state-specific (combinations of duck and pond abundance) harvest policies estimated (a) by the genetic algorithm (b) and by random search for the S_AR_S model of mallard population dynamics.

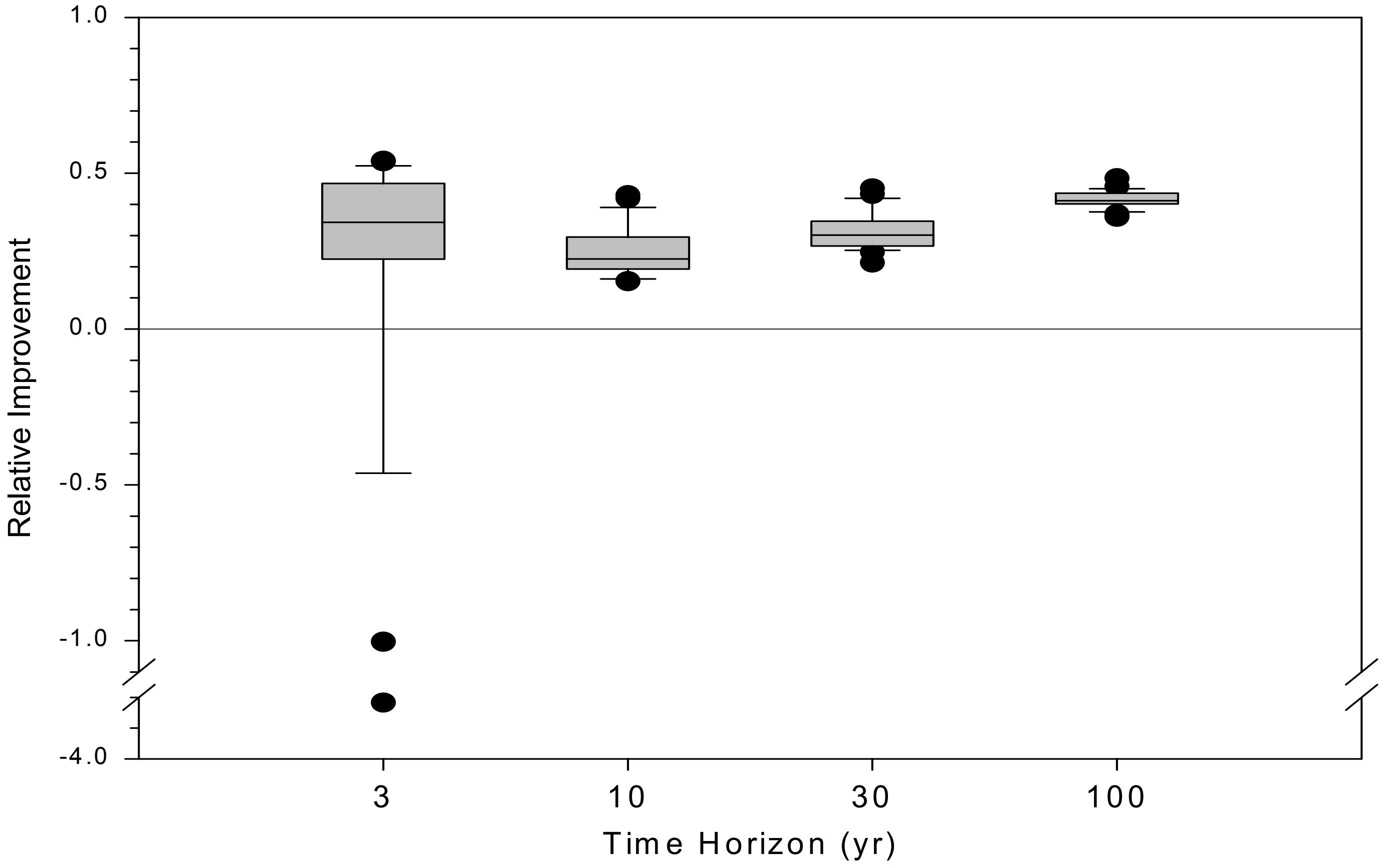
Fig. 3. Expected value of cumulative harvest for the GA-derived harvest policy, expressed relative to the optimal expected value ($R_T^{(1)}$), for 5 simulated time horizons. Each box represents simulation results for 20 combinations of initial mallard population size and pond abundance. Optimal harvest is indicated by the solid reference line; broken reference lines indicate 98 and 99.5 percentiles of optimal harvest.

Fig. 4. Difference in expected cumulative harvest between the GA harvest policy and one derived by random search, expressed relative to the difference between the optimal value and the value provided by random search ($R_T^{(2)}$), for 4 simulated time horizons. Each box represents simulation results for 20 combinations of initial mallard population size and pond abundance. The solid reference line indicates equal proficiency of the GA and random search harvest policies.


Initial State (millions of ducks, millions of ponds)
Harvest Rate
Model ScRs
Model ScRw
Model SaRs
Model SaRw
(12, 6)
(12, 2)
(10, 6)
(10, 2)
(8, 6)
(8, 2)
(6, 6)
(6, 2)
(4, 6)
(4, 2)
0.0
0.1
0.2
0.3
0.4
0.5


A
B
Difference
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
-0.02
-0.04
-0.06
-0.08
-0.10
-0.12
-0.14
Ducks (millions)
2
4
6
8
10
12
Ponds (millions)
1
2
3
4
5
6
7
8


GA/DP Objective value ratio
1.35
1.30
1.25
1.20
1.15
1.10
1.05
1.00
0.95
0.90
0.85
1
3
10
30
100
Time Horizon (yr)


1.0
0.5
0.0
-0.5
-1.0
-4.0
Relative Improvement
3
10
30
100
Time Horizon (yr)