

## Research Article

# Selection and Penalty Strategies for Genetic Algorithms Designed to Solve Spatial Forest Planning Problems

Matthew P. Thompson,<sup>1</sup> Jeff D. Hamann,<sup>2</sup> and John Sessions<sup>1</sup>

<sup>1</sup>Department of Forest Engineering, Resources and Management, Oregon State University, Corvallis, OR 97331, USA

<sup>2</sup>Forest Informatics, Corvallis, OR 97339, USA

Correspondence should be addressed to Matthew P. Thompson, matthew.thompson@oregonstate.edu

Received 12 September 2008; Accepted 24 February 2009

Recommended by Pete Bettinger

Genetic algorithms (GAs) have demonstrated success in solving spatial forest planning problems. We present an adaptive GA that incorporates population-level statistics to dynamically update penalty functions, a process analogous to strategic oscillation from the tabu search literature. We also explore performance of various selection strategies. The GA identified feasible solutions within 96%, 98%, and 93% of a nonspatial relaxed upper bound calculated for landscapes of 100, 500, and 1000 units, respectively. The problem solved includes forest structure constraints limiting harvest opening sizes and requiring minimally sized patches of mature forest. Results suggest that the dynamic penalty strategy is superior to the more standard static penalty implementation. Results also suggest that tournament selection can be superior to the more standard implementation of proportional selection for smaller problems, but becomes susceptible to premature convergence as problem size increases. It is therefore important to balance selection pressure with appropriate disruption. We conclude that integrating intelligent search strategies into the context of genetic algorithms can yield improvements and should be investigated for future use in spatial planning with ecological goals.

Copyright © 2009 Matthew P. Thompson et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

## 1. Introduction

Forest planning is becoming increasingly complicated as managers and decision-makers are increasingly incorporating explicit spatial representations of their planning problems [1]. Thus the design and improvement of solution techniques remain an important avenue of research (e.g., multiple objectives [2–4], even-flow constraints [5, 6], creation/retention of contiguous mature forest patches [7–10], maintenance of a shifting mosaic of age classes across the landscape [11]). For instance, development of solution techniques for harvest scheduling problems where the maximum opening size is limited has received a great deal of attention in the forest planning literature (e.g., [12–23]). Though recent advancements in exact methods permit the solution of some moderate to large instances of area-restricted models [24, 25], in practice heuristic approaches are more common, especially for problems with spatial concerns beyond harvest openings [26]. Thus there remains interest in the use of heuristic algorithms to quickly

generate feasible, high-quality solutions to more complex problems with economic and ecological goals.

One area of interest in spatial forest planning is facilitating sustainable management under the auspices of certification programs (e.g., [25]) and/or forest practice regulations (e.g., [27–30]). Heuristics such as simulated annealing (SA), tabu search (TS), and genetic algorithms (GAs) have been successfully applied to solve complex, spatial forest planning problems [1]. In particular, GAs have shown promise for spatial forest planning problems [4, 31–37]; use of GA has also proven successful for nonspatial forest planning [6, 38]. Other algorithms similarly based on evolutionary concepts have also been demonstrated to perform well (e.g., [39, 40]). The success of GA can in part be attributed to the diversity of solution space that is explored by searching neighborhoods of an entire population rather than of a single solution. Previous research has emphasized the importance of promoting diversity in the search process [22] and demonstrated the importance of using complicated

neighborhood structures in spatial forest planning [16, 41, 42].

Genetic algorithms are stochastic optimization procedures that mimic biological evolution, in particular the concept of natural selection [43]. Candidate solutions are modeled as members of a population competing for survival based upon fitness values (objective functions). Over generations (iterations) it is expected that the population will evolve to be composed of more fit individuals, ideally containing the global optimum. The canonical genetic algorithm [44] mimics evolution via three basic operations: selection, recombination, and mutation. With selection, chromosomes are preferentially chosen to enter the mating pool based upon their fitness values. Once in the mating pool, solutions are stochastically perturbed via recombination and mutation, then passed on to the next generation.

Selection can exert disproportionate influence on algorithm performance because it biases the search for high-fitness solutions and drives convergence of the problem [45]. Absent other evolutionary operators, use of selection alone would eventually result in the complete takeover of the population by a dominant individual. The time required for takeover reflects the selection pressure of the particular selection mechanism used. Higher selection pressures result in lower takeover times, and vice versa. Critical to performance therefore is the selection pressure: if pressure is too high premature convergence to a local optimum becomes likely, and if pressure is too low convergence may prove slow [46].

Beyond identification of appropriate evolutionary mechanisms, another important decision in the design of a GA (or any other heuristic) is how to handle infeasible solutions. One option is to constrain the search by requiring feasibility as a precondition for population membership. Crowe and Nelson [22] however found that retaining feasibility can severely limit neighborhood searches, and Hertz and Widmer [47] argued that penalty functions are necessary to promote diversity in highly constrained combinatorial optimization problems. Lockwood and Moore [21] paired a combinatorial heuristic (SA) with penalty functions to generate feasible solutions for harvest scheduling with spatial constraints. More recently, Falcão and Borges [34] applied penalty values for forest planning problems using a GA. Others to use penalty constraints in forest planning applications, though not in all cases for spatial constraints, include Richards and Gunn [48, 49], Falcão and Borges [38], Boston and Bettinger [18], and Brumelle et al. [50]. Falcão and Borges [38] recommended future research on the use of dynamic penalty functions within a GA.

In this paper we explore the performance of various selection and penalty strategies. In particular we focus on tournament selection, in which solutions directly compete to enter the mating pool. By changing the number of competitors per tournament we are able to experiment with varying levels of selection pressure. Additionally, we experiment with static and dynamic penalty functions. In the latter method, population-level information is used to dynamically update penalty function parameters in order to oscillate around constraint boundaries.

We consider three landscapes of increasing size to examine possible trends between selection pressure, penalty method, and problem complexity. For our analysis we consider a planning environment in which the landowner wishes to achieve certification, the standards for which are roughly representative of the Pacific Coast Standards, Forest Stewardship Council [51]. To seek optimal plans that maximize net present value while retaining contiguous patches of mature forest and limiting harvest area sizes, we employ a variety of GA manifestations. Our decision to investigate modifying the GA is motivated by documented improvements in solution quality obtained from thoughtful modification to other canonical metaheuristic frameworks [8, 13, 34, 38, 49].

Below we present the problem formulation and describe our algorithmic and experimental designs, focusing on selection and penalty strategies. We then present experimental results, discuss algorithmic performance, and offer suggestions for future research.

## 2. Problem Formulation

The formulation we present below draws from the work of Martins et al. [7], Murray et al. [25], and Caro et al. [8]. The work of Murray et al. [25] in turn draws from the seminal work of Goycoolea et al. [24], who show that under certain circumstances a priori enumeration of cutting blocks (and old-growth patches) enables efficient computation through the use of cluster-packing formulations. These formulations constitute extensions to the area-restricted model, wherein smaller planning units can be aggregated into larger contiguous cutting blocks [17]. Of course, with a heuristic framework enumeration of feasible blocks is not necessary, but we opt to present the state-of-the-art mathematical formulation. Our heuristic instead uses a depth-first search technique to determine whether contiguous harvested/unharvested blocks are feasible with respect to opening size/old-growth patch constraints. This enables far larger problems to be solved; Murray et al. [25] state that computing capacities limited their method to enumeration of blocks including at most eight harvest units. Martins et al. [7] note that, for problems with opening size and old-growth constraints, the number of variables increases exponentially with the number of stands, precluding the use of exact methods. This sentiment is echoed by Weintraub and Murray [26], who state that for problems with old-growth patches heuristic approaches are required.

The model parameters and variables are defined as

*CB*: cutting block, a set of contiguous harvest units whose total area is less than the maximum opening size;

*i*: index of harvest units;

*l, j*: index of cutting blocks;

*r*: index of old growth patches;

*t, u*: index of planning periods;

*L*: set of cutting blocks whose total area is less than the maximum opening size;

$R$ : set of old growth patches whose total area is greater than the minimum patch size;

$N_l$ : set of cutting blocks adjacent to cutting block  $l$ ;

$\Omega_l$ : set of cutting blocks incompatible with cutting block  $l$ , defined as

$$\Omega_l = [j \mid (CB_l \cap CB_j \neq \emptyset) \cup j \in N_l]; \quad (1)$$

$S_i$ : set of cutting blocks containing unit  $i$ ;

$OG_{it}$ : set of old growth patches containing unit  $i$  in period  $t$ ;

$p_{it}$ : net present value of harvesting unit  $i$  in period  $t$ ;

$\alpha_l$ : area of cutting block  $l$ ;

$a_r$ : area of old growth patch  $r$ ;

$\bar{A}$ : maximum average opening size;

$MO$ : minimum allowable acreage assigned to old growth status;

$X_{lt}$ : binary variable indicating whether cutting block  $l \in L$  is harvested in period  $t$ ;

$Y_{rt}$ : binary variable indicating whether contiguous patch  $r \in R$  is assigned old-growth status in period  $t$ ;

$Z$ : objective function value (net present value)

$$\max Z = \sum_t \sum_l \left( \sum_{i \in CB_l} p_{it} \right) X_{lt}. \quad (2)$$

Subject to

$$\sum_t \sum_{l \in S_i} X_{lt} \leq 1, \quad \forall i, \quad (3)$$

$$\sum_{u=1}^t \sum_{l \in S_i} X_{lu} + \sum_{r \in OG_{it}} Y_{rt} \leq 1, \quad \forall i, t, \quad (4)$$

$$X_{lt} + X_{jt} \leq 1, \quad \forall t, l, j \in \Omega_l, \quad (5)$$

$$\sum_l (\bar{A} - \alpha_l) X_{lt} \geq 0, \quad \forall t, \quad (6)$$

$$\sum_r a_r Y_{rt} \geq MO, \quad \forall t, \quad (7)$$

$$\begin{aligned} X_{lt} &\in \{0, 1\} \quad \forall l, \\ Y_{rt} &\in \{0, 1\} \quad \forall r, \forall t. \end{aligned} \quad (8)$$

Equation (2) presents the objective function to maximize total net present value from harvest of feasible cutting blocks. Constraints (3) ensure that no unit can be harvested more than once over the planning horizon. Constraints (4) ensure that a unit cannot be included in an old-growth patch if it has already been harvested and limit a unit to membership in at most one harvest block or old-growth patch. Constraints (5) prevent the harvest of incompatible cutting blocks in any period. Incompatible cutting blocks are those blocks that either share a common unit or contain adjacent units, which if harvested in the same period might violate opening size limits (see definition of  $\Omega$ ). If not, their entirety would be accounted for in a separate feasible cutting

block, and Constraints (3) ensure that at most one of those cutting blocks would be scheduled for harvest. If we assume the green-up length is our period length, this formulation satisfies any green-up constraints as well. Constraints (6) limit the average opening size; here we diverge from the Murray et al. [25] formulation by limiting the average opening size by period, not across periods. Constraints (7) ensure that the total area of all feasible old-growth patches exceeds the minimum size requirement. Lastly, Constraints (8) ensure that the decision variables remain binary.

### 3. Algorithm Design

In the following subsections we present our algorithmic design choices and explain their conceptual foundations. Unless otherwise stated, much of this review stems from three excellent sources: Osyczka [52], Bäck et al. [53], and Dumitrescu et al. [45].

**3.1. Selection Strategies.** There are three standard selection schemes: proportional, ranking, and tournament selection. In proportional selection, the most common scheme, individuals are chosen to enter the mating pool according to a probability that reflects an individual's relative fitness. Often this entails simply calculating the ratio of an individual's fitness to the sum total of the population. Proportional selection can be plagued by convergence issues. At one end of the spectrum is premature convergence, which could occur where a single highly fit individual dominates selection into the mating pool. Conversely, slow convergence could occur when a population is sufficiently homogenous to result in minimal selection pressure. To avoid convergence issues a scaling transformation can be used. Scaling transformations modify fitness values according to a user-defined function; common forms include linear, power law, and logarithmic scaling. For minimization problems or for problems where fitness values may be negative, such as when employing penalty functions, scaling is also necessary. The transformation mechanism can be static (independent of generation) or dynamic.

Ranking selection schemes also assign selection probabilities, but based on a chromosome's rank ordering of fitness rather than the actual (or scaled) fitness value. Relative to proportional selection, ranking selection reduces selection pressure when fitness variance is high and increases selection pressure when the variance is low. Ranking schemes can avoid premature convergence and eliminate the need to scale fitness values, but can be computationally expensive because of the need to sort populations.

Once selection probabilities have been assigned, some sampling method is required to populate the mating pool. Most common is a method known as Roulette Wheel Selection (RWS), where in order to select  $n$  possible chromosomes,  $n$  spins of a biased roulette wheel are made. An alternate method is Stochastic Universal Sampling (SUS), where the biased roulette wheel is spun only once. SUS is associated with increased efficiency and reduced variance in the number of offspring assigned to individuals. Yet a third

method, RWS/Uniform, selects one parent using the roulette wheel, which is then paired with a second parent selected according to a uniform distribution [44]. The sampling approaches SUS, RWS, and RWS/Uniform are suitable for proportional and ranking selection methods.

In tournament selection, individuals are randomly chosen to directly compete for entry into the mating pool. The number of individuals competing in each tournament is referred to as  $q$  ( $q$ -ary tournament) and  $q$  is commonly set to 2 (binary tournament). Selection pressure increases as  $q$  increases. Tournaments can be either deterministic, in which the superior solution is always selected, or stochastic, in which inferior solutions may be probabilistically chosen over superior alternatives. A special case of the latter is a Boltzmann tournament, which has clear similarities to simulated annealing. Advantages to tournament selection include efficient time complexity, especially if implemented in parallel, low susceptibility to takeover by dominant individuals, and no requirement for fitness scaling or sorting. Tournament selection is thought to accelerate the process of evolution and may yield better solutions.

Goldberg and Deb [54] performed a comparative analysis of the selection schemes we are addressing. They found that ranking and tournament selection outperformed proportional selection in terms of maintaining steady pressure toward convergence. They further demonstrated that linear ranking selection and stochastic binary tournament selection have identical expectations, but recommended binary tournament selection because of its more efficient time complexity. Per generation, proportional selection with RWS has a time complexity of  $O(n^2)$ , though that can be reduced to  $O(n \log n)$  using a binary search mechanism, where  $n$  refers to the population size and  $O()$  the time complexity of the particular selection algorithm. Use of the SUS sampling scheme can further reduce proportional selection time complexity to  $O(n)$ . Rank-based selection is  $O(n \log n)$  to sort plus the selection time, which can range from  $O(n)$  to  $O(n^2)$ . Lastly, tournament selection has time complexity  $O(n)$ .

In addition to multiple selection schemes there exist multiple metastrategies for the selection operator. These include decisions of whether to employ elitism and generational versus nongenerational models. With elitism the best individual(s) from each generation are kept to be passed on to future generations. In some respects elitism is analogous to the intensification phase of TS. By passing on a high-quality solution to the next generation, elitism ensures further search of that solution and its neighbors. In generational selection a newly created population replaces that which created it, as opposed to limiting the number of recombinations and replacements per cycle. The generational model is considered more appropriate for optimization [55]. Use of elitism or nongenerational models necessitates selection of an appropriate replacement strategy, which itself has numerous options, such as replacement of the worst and replacement of the oldest.

Within the forest planning literature most applications have used some form of proportional or rank-based selection. Table 1 presents a nonexhaustive review of selection

schemes used in forest planning applications. “Modified RWS” refers to what essentially amounts to a scaling mechanism developed by Falcão and Borges [38] to reduce the probability of selecting less fit individuals. In general, most applications employ elitism and use the generational model. We found that few authors investigated the performance of alternate selection schemes. Some present instances of GAs that include functionality for multiple selection schemes but do not report comparative results (e.g., [36, 56]). Interestingly, few authors considered using SUS to speed up the process of probability-based selection methods, even though some commented on the slow performance of GA.

During preliminary experimentation we examined a wide range of possible selection strategies and compared solution quality and time. We tested our model formulation on a small 50-unit landscape adapted from the Daniel Pickett Forest Problem [57]. At this stage we were more interested in identifying the relative merits of the various selection strategies than in validating the heuristic. Results indicated that tournament selection was the superior strategy, achieving the highest solution qualities with low computing times. There emerged a clear pattern of improving performance with increasing competitors per tournament. Rank-based selection also achieved high-quality solutions, but at significantly larger computing times, which we attribute to the need to sort the population each generation. These findings agreed with that of Goldberg and Deb [54].

After analyzing the initial results from the small 50-unit landscape we identified proportional selection and deterministic tournament (DT) selection for additional experimentation. We consider proportional selection to represent the more standard implementation as used by researchers (see Table 1). We opted for the SUS sampling scheme in recognition of the possible computational improvements. Our rationale for selecting DT schemes was threefold: (1) DT outperformed ranking selection methods in terms of solution quality and computational effort, (2) DT yielded the best absolute objective function values, and (3) DT did not require identification of additional model parameters (initial temperature and cooling rate) required for a stochastic tournament.

In total we developed four tournament strategies for experimentation, two of which were static and two dynamic. The static selection schemes employ different tournament sizes:  $q = 2$  (binary) and  $q = 7$ . In the first dynamic scheme  $q$  is randomly assigned each generation, thereby creating a search trajectory with fluctuating selection pressure. This is similar in some respects to use of random tabu tenure (e.g., [8]). We sampled  $q$  from a uniform distribution over the range [2,10]. The second dynamic scheme attempts to accelerate evolution by increasing selection pressure over time. We implemented a simple rule whereby  $q$  switches from 2 to 7 after the generation count exceeds one-third of the maximum allowable generations. If the size of  $q$  was instead updated according to population-based information, this technique would be similar to reactive tabu search, wherein the tabu list size is dynamically updated [48, 58].

All selection strategies we used were elitist and generational. The size of the elitism archive was set to one,

TABLE 1: Various selection schemes implemented in the forest planning literature. RWS stands for roulette wheel sampling.

Scheme	Details	Source
Proportional	RWS	Palahí et al. [33]; Boston and Bettinger [30]
	Modified RWS	Falcão and Borges [34, 38]
	RWS/uniform	Bettinger et al. [42]
Rank	RWS	Dewey et al. [56]; Lu and Eriksson [35]
	RWS/uniform	Pukkala and Kurttila [31]
Tournament	Binary, deterministic	Ducheyne et al. [6]

meaning only the best solution was retained to be passed to the next generation. Our replacement strategy removed the worst chromosome in the population and in its place inserted the solution from the elitist archive. We also employed an elitist feasibility archive, containing the best feasible solution found. If at the end of the run the best overall solution found was not feasible, the solution from the feasibility archive was reported instead. This ensures all runs output feasible, high-quality solutions.

**3.2. Penalty Functions.** Use of penalty functions requires slight modifications to the mathematical formulation as presented above. Let  $x^0$  represent a candidate solution, and let  $g_{kt}(x^0)$  represent constraint  $k$  in period  $t$ . Reformulate the constraints to the form  $g_{kt}(x^0) \leq 0$ , so that a positive value indicates infeasibility. Now define a transformation function  $h(x^0)$  as in (9), and further define some penalty function  $\theta(h(x^0))$ . Whenever constraints are violated ( $g(x^0) > 0$ ) the penalty function is invoked and the objective function reduced accordingly. Equation (10) presents the new objective function:

$$h(x^0) = \begin{cases} g(x^0) & \text{if } g(x^0) > 0, \\ 0 & \text{otherwise,} \end{cases} \quad (9)$$

$$\max Z' = \sum_t \sum_l \left( \sum_{i \in l} p_{it} \right) X_{lt} - \sum_k \sum_t \theta_k(h_{kt}(x^0)). \quad (10)$$

The penalty function  $\theta(h(x^0))$  is a simple linear function of the magnitude of the constraint violation (11). In the static penalty approach  $c_j$  was set to roughly 5% of the estimated trivial upper bound (TUB) using methods described below (see Section 4):

$$\theta_k(h(x^0)) = c_k h_{kt}(x^0), \quad \forall j, t. \quad (11)$$

In the dynamic penalty approach the penalty parameter  $c_k$  varies across generations. Our parameter updating scheme is an adaptation of a TS concept known as strategic oscillation [48, 49, 59, 60], where penalty term coefficients are dynamically adjusted according to running tallies on solution feasibility. The penalty term  $c_k$  is initially set to some upper bound, then halved every time  $M_k$  consecutive solutions have been feasible with respect to constraint  $k$ , or doubled after  $M_k$  consecutive infeasible solutions. Of course, the choice of  $M_k$  can affect algorithm performance, and too small a value may hinder diversification. Gendreau

et al. [59] however reported that their tabu search algorithm was relatively insensitive to the value of  $M_k$ . The intent of strategic oscillation is to control the direction of search so that the solution space examined moves in and out of feasibility near constraint boundaries to facilitate exploration of active constraints.

For the population-based GA we instead update penalty parameters according to running tallies incremented when a certain percentage of the population is feasible. This requires identification of an additional parameter:  $Pr_k$ , the threshold for the proportion of the population that is feasible with respect to constraint  $k$ , beyond which  $M_k$  is updated. In our implementation we arrived at values of 25 for  $M_k$ , and 50% for  $Pr_k$  after initial experimentation. The upper bound for  $c_k$  was set to roughly 10% of the estimated TUB. After penalties were levied some fitness values became negative, so it was necessary to scale solutions in order to apply proportional selection. We opted for a dynamic scaling mechanism, wherein the entire populations' fitness values are increased according to the fitness of the worst individual. Algorithm 1 displays the general outline of this dynamic penalty method, plus an overview of the genetic algorithm.

**3.3. Diversity Preserving Mechanisms.** Recombination and mutation promote diversity in the solution space, serving to counterbalance selection pressure and avoid premature convergence. With recombination, solutions in the mating pool are probabilistically assigned to be either transferred into the next generation as is, or to mate with another solution. Mating swaps genes between parents to generate two new offsprings in a process called crossover. Mutation alters solutions by randomly changing alleles to values that may not have been possible through inheritance alone. Traditionally recombination has been the primary driver of solution disruption, with mutation playing more of a background role. Though disruption is an important process, too much random perturbations can effectively result in random search and may lead to poorer performance. Because of our focus on selection, we kept our exploration of recombination and mutation strategies to a minimum. To avoid excessive disruption from recombination we only employed 1-point crossover recombination, although with a relatively high mating probability of 60% and a mutation rate of 1%. As with issues relating to selection and penalty strategies, these values were arrived at after initial experimentation.

```

Canonical Genetic Algorithm
 $P^0 \leftarrow \text{Initialize\_Population}()$ 
Evaluate\_Fitness( $P^0$ )
While (! Termination\_Criteria)
   $P^1 \leftarrow \text{Selection}(P^0)$ 
   $P^2 \leftarrow \text{Recombination}(P^1)$ 
   $P^0 \leftarrow \text{Mutation}(P^2)$ 
  Evaluate\_Fitness( $P^0$ )

Evaluate\_Fitness( $P$ )
Calculate\_NPV( $P$ )
Calculate\_Penalty( $P$ )

Calculate\_Penalty( $P$ )
For (All constraints  $k$ )
  If (Percent\_Infeas( $k$ )  $\geq$   $\text{Pr}_k$ )
    Infeas\_Count++
    Feas\_Count = 0
  Else
    Infeas\_Count = 0
    Feas\_Count++
  If (Infeas\_Count  $\geq$   $M_k$ )
     $c_k = c_k * 2$ 
  If (Feas\_Count  $\geq$   $M_k$ )
     $c_k = c_k * 1/2$ 
For (All solutions  $x$  in  $P$ )
   $\theta(h(x)) = c_k h(x)$ 

```

ALGORITHM 1: Pseudocode for the genetic algorithm, plus fitness and penalty evaluation functions. The canonical genetic algorithm mimics evolution via three basic operations: selection, recombination, and mutation. Fitness evaluation consists of calculating net present value and then levying penalties. This algorithm displays the dynamic penalty approach, which is modeled after strategic oscillation. For each constraint  $k$ , the penalty coefficient  $c_k$  is increased (decreased) according to a running tally on the number of consecutive generations in which the population is infeasible (feasible). For the static penalty approach, the penalty coefficient  $c_k$  remains constant across generations.

**3.4. Chromosome Representation.** Solutions are represented as chromosomes, which are comprised of genes, the biologic analog for decision variables. Alleles are the actual values genes may take. Though various methods exist to codify decision variables within the chromosome, a suitable approach is to use integer coding, where the integer value corresponds to the specific management prescriptions assigned to each unit [38]. In our representation, a chromosome consists of  $n$  genes, where  $n$  is the number of harvest units. For a problem with a planning horizon of  $t$  periods, alleles can take values in the range  $[0, t]$ , indicating either a no harvest option or the period in which a unit is harvested. Our heuristic implementation therefore has  $n$  integer decision variables ( $n(t+1)$  binary decision variables), which is significantly less than in the exact formulation above.

**3.5. Population Size.** Population size can significantly impact algorithm performance and is therefore an important choice in design. Too small populations may insufficiently span

the solution space, while larger populations may lead to poor computational efficiency [44]. For instance, Falcão and Borges [38] reported that initial experimentation with populations larger than 50 chromosomes slowed convergence rates without improving performance. Reeves [44] suggested that population sizes as low as 30 chromosomes can be adequate, and in fact this population size was chosen by Pukkala and Kurttila [31] as well as Falcão and Borges [34, 38]. Venema et al. [32] and Dewey et al. [56] used slightly larger populations of 40 and 50 chromosomes, respectively. As with diversity preserving mechanisms, we limited our exploration of population size. To facilitate exploration of the solution space while limiting computational effort we opted for a modestly large population of 50 chromosomes, kept constant across problem size. As with other decisions in heuristic design, this choice reflects a compromise between quality and computational effort.

**3.6. Stopping Criterion.** Though it is generally expected that with additional generations solution quality improves, marginal gains decrease and may eventually asymptote [38]. Typically genetic algorithms terminate after a predetermined number of generations have passed (e.g., [31]) or after a sequence of consecutive generations without objective function improvement (e.g., [32, 42]). Alternatively, the algorithm can terminate after the population is sufficiently homogenized, as measured by objective function variance. Due to the varying pressures of the selection strategies employed we would expect populations to converge at different times (generation counts) for different strategies. As we were interested in the temporal (generational) performance of the various selection strategies, we opted to use a maximum generation count as a stopping criterion. Looking at the strategies' performance at different points in time enables us to compare convergence properties from a common perspective.

## 4. Experimental Design

We examined heuristic performance across three simulated landscapes of increasing size (100, 500, 1000 units), over 5 planning periods of length 5 years. We generated landscapes to obtain complete datasets (e.g., all polygons contained fixed and known values), rather than using operational datasets for our comparative study. For each landscape, we randomly generated polygon centers (e.g., landings) in  $\mathbb{R}^2$ . We then used the `deldir` () package [61] of the *R* statistical software [62] to obtain a set of convex polygons from the resulting Voronoi diagrams [63]. Current age classes were assigned to each polygon using a uniform random sample. The uniformity of the age class is thought to be representative of a fully regulated ownership, now transitioning to a distribution comprised of more mature forest to achieve certification. Finally, current and future conditions for each treatment unit were defined using the yield function similar to the growth function developed by [64]:  $V(y) = \sum_{i=0}^y e^{2.9 \cdot \log(i) - 0.001i^2}$ , where  $y$  is the age, in periods, of the stand since planting and  $V(y)$  are mbf/acre.

Table 2 presents the details of the simulated landscapes. In all instances the maximum opening size was set to 48.56 ha (120 ac), the maximum average opening size to 32.37 ha (80 ac), the minimum acreage for an eligible contiguous old-growth patch 8.09 ha (20 ac), and the planning horizon extended 5 periods. Forest units were considered eligible for recruitment into old-growth patches if they were at least 80 years old. MO represents the minimum acreage in old-growth patches across the landscape, which varied with landscape. TUB stands for the trivial upper bound, which can be estimated using the methods outlined in Caro et al. [8]. In the linear program relaxation, harvest decisions for each stand remain binary and constraints ensure a minimal total acreage in mature forest patches, but no spatial relationships for opening size or patch size are included.

For each selection/penalty pairing we ran the GA 50, 30, and 20 times for landscapes A, B, and C, respectively. Opting for less runs on the larger landscapes reflects a compromise between computational effort and confidence in results. All initial solutions in the population were generated randomly. We opted to compare selection/penalty pairing at two different generation counts, which varied with landscape. For landscape A we compared strategies after 750 and 1500 generations. The maximum generation counts increased to 1500/3000 and 1500/6000 for landscapes B and C, respectively. The increase in the maximum generation count is not proportional to the increase in problem size, reflecting a compromise between solution quality and computational effort. After identifying a superior selection/penalty pairing we then raise the maximum generation count to 10 000 for a single run.

The final stage of the experimental design is identification of appropriate validation methods. Bettinger et al. [65] identify six levels of validation for heuristics, which range from no validation to comparison with known optima. Level 2, which includes establishing average performance and which we present here, is encouraged in all applications. Level 3 validation, which we also perform, compares the proposed heuristic's performance to other standard heuristics. As stated earlier, we consider proportional selection as the standard implementation of GA in forest planning and compare its performance to various tournament strategies. Lastly, as Level 6 (known as optimal comparison) was infeasible due to problem size [7, 26], we opted for Level 5 validation by comparing our results to relaxed LP solutions (TUB).

We solved the TUB problems to within 0.00001% of optimal using the GAMS software and the CPLEX MIP solver. We processed the data and generated the GAMS input files using Microsoft Visual Studio and Visual Basic for Applications. We developed our GA in Microsoft Visual C++ 6.0 and generated results on a computer with a dual-core 2.2 GHz processor and 2 GB DDR2 SDRAM.

## 5. Results

Computation time was comparable between strategies for each landscape, with proportional selection on average

slightly longer than tournament selection, but not significantly so. Solution time increased dramatically with problem size. Average solution times for landscape A were 10 seconds (750 generations) and 18 seconds (1500 generations). For landscape B average times increased to 627 seconds (1500 generations) and 1326 seconds (3000 generations) seconds, and further to 2695 seconds (1500 generations) and 10 044 seconds (6000 generations) for landscape C. These roughly correspond to 83.33, 2.26, and 0.60 generations per second for landscapes A, B, and C, respectively. Thus, a 10-fold increase in integer decision variables led to a more than 100-fold increase in computing time per generation. Had we opted to increase population size with landscape size, we expect this ratio would be much greater.

Our GA was able to identify feasible solutions within 94.82%, 90.91%, and 89.65% of the estimated TUB for landscapes A, B, and C, respectively. To reiterate, the TUB is calculated without spatial constraints limiting the size of harvest openings and mature forest patches. Similar results (in terms of solution quality) were reported by Crowe and Nelson [22] when using simulated annealing to solve the area-restricted model and by Caro et al. [8] for a more ecologically focused harvest schedule solved with tabu search. For all landscapes the algorithm was able to consistently identify feasible solutions.

Figures 1, 2, 3, 4, 5, and 6 display box plots of the various selection/penalty strategies applied to each test instance (landscape + generation count). Static and dynamic penalty strategies for the same selection strategy are displayed side by side (static on the left side). Reading from left to right, the first selection strategy presented is proportional selection and the rest are deterministic tournaments with various competitor pool sizes ( $q$ ). The label " $q = U$ " corresponds to the strategy wherein  $q$  was sampled from a uniform distribution over the range (2, 10), and the label " $q = 2-7$ " corresponds to the strategy where  $q$  increased from 2 to 7 during the search. Tables 3 and 4 compare results for the various selection/penalty strategies on the basis of best (Table 3) and average (Table 4) solution quality, expressed in percentage of the estimated trivial upper bound (TUB).

The best performing selection/penalty pairings varied with landscapes and with the maximum generation count. Across landscapes the dynamic penalty method achieved the highest overall and average solution quality. Behavior for landscape A across generation counts was generally consistent, with proportional/dynamic superior in both cases (750 and 1500 generations). The top three selection/penalty strategies for landscape A (1500 generations), in order, were proportional/dynamic (94.82%),  $q = U$ /static (93.07%), and  $q = 7$ /static (93.06%). For landscapes B and C after only 1500 generations tournament selection outperformed proportional selection, independent of implementation. After letting the search continue for additional generations however proportional selection's relative performance improved. The top three strategies for landscape B (3000 generations), in order, were  $q = 7$ /dynamic (90.91%),  $q = 2-7$ /dynamic (90.50%), and proportional/dynamic (90.42%). On landscape C,

TABLE 2: Description of simulated landscapes; MO stands for minimal acres required to be in old-growth patches of a minimum size and TUB represents the nonspatial trivial upper bound as estimated using GAMS.

Landscape	Total area (ha)	Units	MO (ha)	TUB (\$)
A	202.34 (500 ac)	100	40.47 (100 ac)	857077.28
B	1011.71 (2500 ac)	500	202.34 (500 ac)	3663275.48
C	9995.74 (24,700 ac)	1000	2023.43 (5000 ac)	37808102.09

TABLE 3: Best solutions for landscapes A, B, and C. We present the best solution in terms of percent of trivial upper bound (TUB), for each selection/penalty combination. The secondary column heading represents the maximum number of generations. PROP: Proportional selection; BIN DT: Binary, deterministic tournament; 7 DT: A deterministic tournament with 7 competitors; U(2, 10) DT: A deterministic tournament with the number of competitors sampled from a uniform distribution on the range (2, 10); 2–7 DT: A deterministic tournament where the number of competitors increases from 2 to 7 throughout the course of the search.

Penalty	Selection	Landscape A		Landscape B		Landscape C	
		750	1500	1500	3000	1500	6000
Static	PROP	88.04	92.39	78.66	87.64	68.90	86.74
	BIN DT	87.72	91.83	82.52	88.03	72.97	83.86
	7 DT	86.94	93.06	84.24	89.85	73.43	84.10
	U(2, 10) DT	87.51	93.07	83.66	89.36	73.70	83.82
	2–7 DT	87.67	92.35	83.03	88.71	72.64	83.94
Dynamic	PROP	89.70	94.82	79.84	90.42	70.11	89.65
	BIN DT	85.44	91.89	82.51	89.66	71.97	85.92
	7 DT	87.77	92.73	85.03	90.91	72.69	84.56
	U(2, 10) DT	87.90	91.80	84.10	90.27	72.75	84.33
	2–7 DT	87.03	91.85	85.35	90.50	74.00	84.34

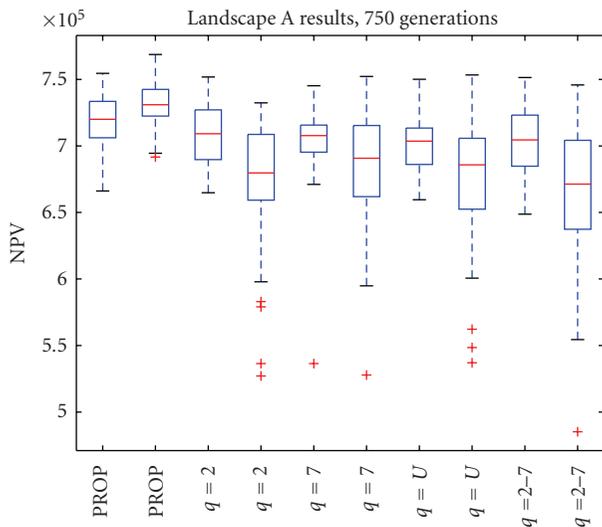


FIGURE 1: Results for Landscape A, 750 generations. This figure presents a box-plot of solution quality (NPV;  $y$ -axis) for each paired selection/penalty strategy tested. The  $x$ -axis displays the strategy; static and dynamic penalty strategies for the same selection strategy are displayed side by side (static on the left side). Reading from left to right, the first selection strategy presented is proportional selection and the rest are deterministic tournaments with various competitor pool sizes ( $q$ ). The label “ $q = U$ ” corresponds to the strategy wherein  $q$  was sampled from a uniform distribution over the range (2, 10), and the label “ $q = 2-7$ ” corresponds to the strategy where  $q$  increased from 2 to 7 during the search.

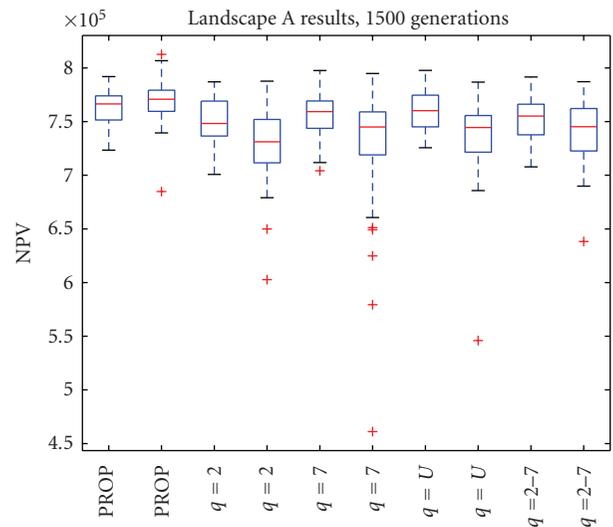


FIGURE 2: Results for Landscape A, 1500 generations. This figure presents a box-plot of solution quality (NPV;  $y$ -axis) for each paired selection/penalty strategy tested. The  $x$ -axis displays the strategy; static and dynamic penalty strategies for the same selection strategy are displayed side by side (static on the left side). Reading from left to right, the first selection strategy presented is proportional selection and the rest are deterministic tournaments with various competitor pool sizes ( $q$ ). The label “ $q = U$ ” corresponds to the strategy wherein  $q$  was sampled from a uniform distribution over the range (2, 10), and the label “ $q = 2-7$ ” corresponds to the strategy where  $q$  increased from 2 to 7 during the search.

TABLE 4: Average solutions for landscapes A, B, and C. We present the average solution in terms of percent of trivial upper bound (TUB), for each selection/penalty combination. The secondary column heading represents the maximum number of generations. PROP: Proportional selection; BIN DT: Binary, deterministic tournament; 7 DT: A deterministic tournament with 7 competitors; U(2, 10) DT: A deterministic tournament with the number of competitors sampled from a uniform distribution on the range (2, 10); 2–7 DT: A deterministic tournament where the number of competitors increases from 2 to 7 throughout the course of the search.

Penalty	Selection	Landscape A		Landscape B		Landscape C	
		750	1500	1500	3000	1500	6000
Static	PROP	83.85	89.13	75.11	85.99	65.08	85.78
	BIN DT	82.61	87.63	80.19	86.84	69.96	82.33
	7 DT	82.11	88.31	80.98	87.86	70.22	81.60
	U(2, 10) DT	81.98	88.71	80.83	87.11	70.94	81.93
	2–7 DT	82.23	87.77	80.59	87.42	70.39	82.28
Dynamic	PROP	85.38	89.78	75.94	86.41	66.83	88.00
	BIN DT	78.68	85.04	79.85	87.86	68.97	83.80
	7 DT	80.05	84.48	81.10	88.31	68.54	82.29
	U(2, 10) DT	78.80	86.03	81.17	88.20	69.41	83.08
	2–7 DT	77.86	86.29	80.87	87.84	69.56	82.48

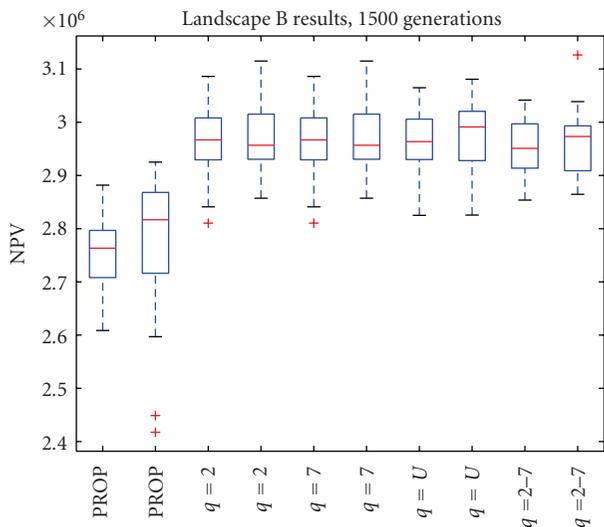


FIGURE 3: Results for Landscape B, 1500 generations. This figure presents a box-plot of solution quality (NPV; y-axis) for each paired selection/penalty strategy tested. The x-axis displays the strategy; static and dynamic penalty strategies for the same selection strategy are displayed side by side (static on the left side). Reading from left to right, the first selection strategy presented is proportional selection and the rest are deterministic tournaments with various competitor pool sizes ( $q$ ). The label “ $q = U$ ” corresponds to the strategy wherein  $q$  was sampled from a uniform distribution over the range (2, 10), and the label “ $q = 2-7$ ” corresponds to the strategy where  $q$  increased from 2 to 7 during the search.

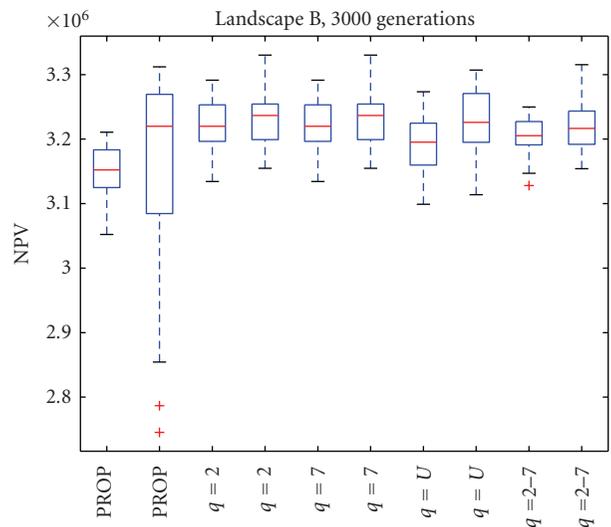


FIGURE 4: Results for Landscape B, 3000 generations. This figure presents a box-plot of solution quality (NPV; y-axis) for each paired selection/penalty strategy tested. The x-axis displays the strategy; static and dynamic penalty strategies for the same selection strategy are displayed side by side (static on the left side). Reading from left to right, the first selection strategy presented is proportional selection and the rest are deterministic tournaments with various competitor pool sizes ( $q$ ). The label “ $q = U$ ” corresponds to the strategy wherein  $q$  was sampled from a uniform distribution over the range (2, 10), and the label “ $q = 2-7$ ” corresponds to the strategy where  $q$  increased from 2 to 7 during the search.

proportional selection clearly outperformed all forms of tournament selection, with the dynamic penalty approach significantly outperforming the static approach. The top three strategies on landscape C were proportional/dynamic (89.65%), proportional/static (86.74%), and binary ( $q = 2$ )/dynamic (85.92%). Considering average performance, the

best selection/penalty pairs for landscapes A (1500 generations), B (3000 generations), and C (6000 generations) were proportional/dynamic (89.78%),  $q = 7$ /dynamic (88.31%), and proportional/dynamic (88.00%), respectively. Based on these results we selected the proportional/dynamic strategy for further experimentation.

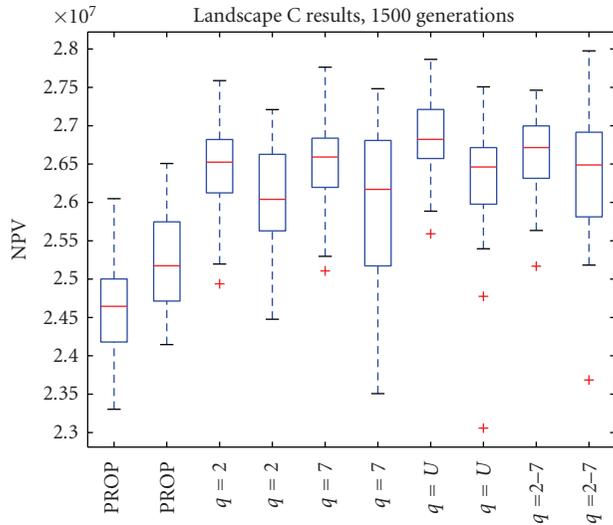


FIGURE 5: Results for Landscape C, 1500 generations. This figure presents a box-plot of solution quality (NPV;  $y$ -axis) for each paired selection/penalty strategy tested. The  $x$ -axis displays the strategy; static and dynamic penalty strategies for the same selection strategy are displayed side by side (static on the left side). Reading from left to right, the first selection strategy presented is proportional selection and the rest are deterministic tournaments with various competitor pool sizes ( $q$ ). The label “ $q = U$ ” corresponds to the strategy wherein  $q$  was sampled from a uniform distribution over the range (2, 10), and the label “ $q = 2-7$ ” corresponds to the strategy where  $q$  increased from 2 to 7 during the search.

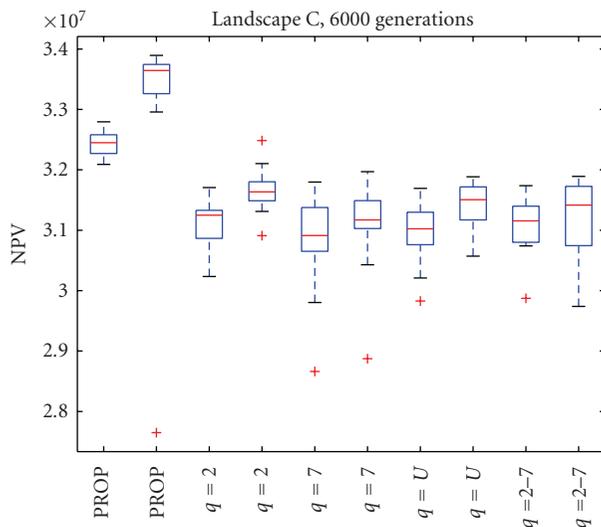


FIGURE 6: Results for Landscape C, 6000 generations. This figure presents a box-plot of solution quality (NPV;  $y$ -axis) for each paired selection/penalty strategy tested. The  $x$ -axis displays the strategy; static and dynamic penalty strategies for the same selection strategy are displayed side by side (static on the left side). Reading from left to right, the first selection strategy presented is proportional selection and the rest are deterministic tournaments with various competitor pool sizes ( $q$ ). The label “ $q = U$ ” corresponds to the strategy wherein  $q$  was sampled from a uniform distribution over the range (2, 10), and the label “ $q = 2-7$ ” corresponds to the strategy where  $q$  increased from 2 to 7 during the search.

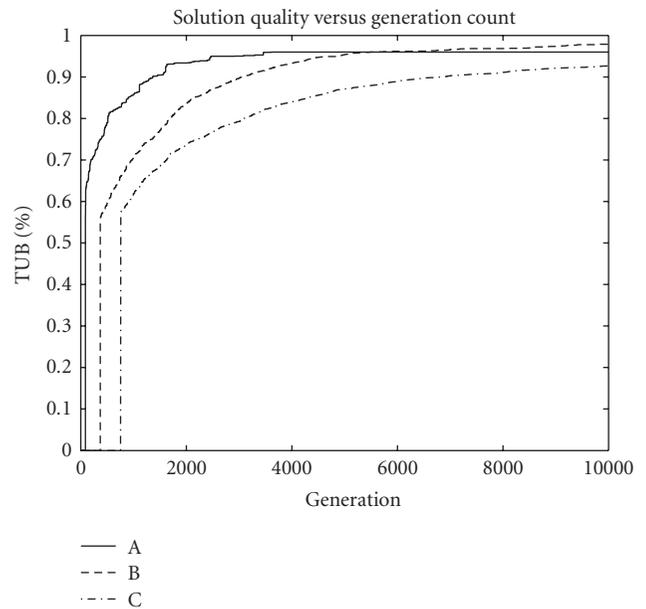


FIGURE 7: Solution Quality versus Generation Count, single run of algorithm over 10 000 generations. Values of zero indicate that the population was comprised of infeasible solutions. All three trajectories were created using proportional selection paired with the dynamic penalty strategy.

Figure 7 graphs solution quality against generation count for the three landscapes, using a single run of the proportional/dynamic strategy. After 10 000 generations our GA was able to find solutions within 95.99% (121 seconds), 97.88% (4752 seconds), and 92.68% (17 388 seconds) of TUB, for landscapes A, B, and C, respectively. The algorithm’s performance reached an asymptote after only 3607 generations for landscape A, but was still improving at 10 000 generations for both landscapes B and C. Extending the evolutionary search from 750 to 1500 generations on landscape A improved solution quality from 82.73% to 90.46% of TUB (9.3% improvement). Extending the search through 10 000 generations improved performance to 95.99% of TUB (6.1% improvement over quality at 1500 generations). With landscape B, solution quality increased from 77.05% to 89.67% of TUB from 1500 to 3000 generations (16.37% improvement), and further to 97.88% (9.16% improvement) after an additional 7000 generations. With landscape C, solution quality increased from 79.25% to 88.92% of TUB from 1500 to 6000 generations (29.66% improvement), and further to 92.68% (4.22% improvement) after an additional 4000 generations.

As stated earlier, a common alternative to a maximum generation count is to cease the evolutionary search after a sequence of nonimproving generations. Using 50 consecutive generations without improvement as the trigger and assessing the results presented in Figure 7, the algorithm would have stopped after 1729, 4741, and 9569 generations for landscapes A, B, and C, respectively. At these generation counts, solution quality was 93.10% (A), 94.91% (B), and 92.33% (C) of TUB. Extending to 100 consecutive generations, search would have ceased after 1890 and 4791 generations

for landscapes A and B, respectively. For Landscape C the algorithm would have continued to search beyond 10 000 generations. Solution quality at these generation counts was 93.37% (A) and 94.91% (B) of TUB.

## 6. Discussion

Analysis of our results indicates the presence of some noteworthy trends. The best overall strategy appears to be proportional selection paired with the dynamic penalty approach. Expectedly, solution quality improves with increasing generation count. For a given generation count, solution quality worsens with increasing problem size, which agrees with other results from application of metaheuristics to spatial forest planning [8, 22]. This indicates that larger problems take more time to solve. Tournament selection initially outperforms proportional selection, but given additional generations proportional selection appears at least as good if not superior, especially in the case of landscape C. This finding is suggestive of premature convergence, and may be attributed to a poor balance between selection pressure and solution disruption. Similarly, within tournament strategies, solution quality appears influenced by an inverse relationship between problem size and selection pressure. As problem size increases selection strategies with lower selection pressure perform better. Compare, for instance, the performances of the binary ( $q = 2$ ) tournament with the  $q = 7$  tournament across the landscapes (Figures 2, 4, and 6).

There is not a compelling reason to opt for dynamic tournament sizing, at least as implemented. This conclusion however could relate more to the selection pressure than the temporal nature of the tournament sizing, as both dynamic strategies ( $q = U$ ,  $q = 2-7$ ) are effectively bounded in terms of competitors per tournament by the two static strategies ( $q = 2$ ,  $q = 7$ ). On average the  $q = U$  strategy has 6 competitors per tournament, so that it performed similarly to the 7 DT strategy is sensible. Why the 2-7 DT did not perform as well is not clear, though in hindsight it might make more sense to try a  $q = 7-2$  approach to reduce selection pressure further into the search and prevent premature convergence.

Our results regarding selection strategies generally disagreed with our own preliminary results. That is, tournament selection did not appear to perform as well as we had expected, relative to proportional selection. One possible explanation is simply the No Free Lunch theorem [66], which states that there is no algorithm that is superior across all problem classes. However, as we stated above, another explanation is an imbalance between selection pressure and solution disruption. The imbalance is related to problem size, so that for the small problem solved initially, tournament schemes with a large number of competitors appeared to perform well. The limited role of recombination and mutation in our experiments may have also contributed to this imbalance, especially for high-pressure tournament strategies ( $q = 7$ ,  $q = U$ ,  $q = 2-7$ ). It may therefore be advisable to pair tournament-based strategies with alternate diversity-preserving mechanisms. One candidate is to

employ a more disruptive recombination operator, such as 2-point or uniform crossover.

Results regarding static versus dynamic penalties are mixed, though on balance the dynamic approach appears superior. For landscape A the static approach yields better solutions (best and average) across tournament strategies, with the opposite being true for landscapes B and C. For proportional selection the dynamic strategy is always dominant. The dynamic approach exhibits greater variability, but from the box plots it is clear that a random draw from the dynamic subpopulation would in most cases have a higher expectation than from the static. The difference between penalty approaches becomes more marked after additional generations, meaning the algorithm can likely identify solutions closer to the edge of the solution space after more time spent intensively searching around the constraint boundaries.

## 7. Conclusions

We presented the design of a dynamic genetic algorithm for spatially constrained, ecologically oriented forest planning problems, and demonstrated that our GA could find high-quality, feasible solutions. Our design approach incorporated principles from tabu search, in particular the ideas of strategic oscillation, reactive search, and random tabu tenure. We applied those ideas in the abstract to update penalty functions and tournament sizing. Results indicate that tournament selection, though theoretically attractive, may become increasingly susceptible to premature convergence as problem size increases. It is therefore important to pair selection strategies with suitable disruptive operators. Despite this apparent flaw, tournament selection was still able to identify high-quality solutions, often very close to or better than solutions obtained by proportional selection. The overall satisfactory performance of the GA is attributed in part to the inherent ability of population-based algorithms to retain diversity in the solution space in the face of increasing selection pressure.

Our results also demonstrated that applying dynamic penalty rates based on population-level metrics can yield improvements relative to a static approach. This represents a novel application of strategic oscillation, seeking to direct the entire population toward constraint boundaries. Proportional selection paired with our dynamic penalty approach appeared the best overall strategy. Conceptually this has merits; low selection pressure may promote the mating of infeasible, penalized solutions near constraint boundaries that might otherwise lose in a deterministic tournament. Applying penalties, static or dynamic, is essentially a scaling transformation, biasing the evolutionary search for feasible solutions.

There are several promising avenues for future research. An idea we promoted here is to pursue exploration of common threads among metaheuristics. Though others have promoted the idea of hybridizing heuristics (e.g., [13]), our approach pairs and adapts concepts rather than iteratively switching between search strategies. Parallel implementation

would dramatically reduce computing times and may render void the common complaint that GAs are slow compared to other heuristics (though the neighborhood search of tabu search can also be implemented in parallel). Tournament selection in particular lends itself well to parallel processing. Of the ideas we introduced, such as dynamic tournament sizing and penalty functions, we only briefly explored possible implementations. Tournament size, for instance, could increase based upon metrics regarding heterogeneity of the population rather than a simple generational count. Dynamic recombination and mutation could also be pursued and could be modified in concert with tournament sizing to balance selection pressure and disruption throughout the search process.

It is possible that alternate implementations of the dynamic approach could yield more improvements. One suggestion is to tailor the updating of penalty parameters to the specific constraints. In our experience the maximum opening size and old-growth constraints were by far the most limiting. Thus it may not be appropriate to trigger penalty updating according to the same rubric for all sets of constraints; experimenting with different values for  $Pr_j$  and  $M_j$  may yield improved results. Additionally, after a sequence of feasible generations penalty parameters drop near zero, resulting in a subsequent sequence of infeasible generations. It might make sense to instead employ floor functions for penalty parameters for particularly difficult constraints to still encourage searching around constraint boundaries while trying to avoid prolonged periods of infeasibility. Other functional forms for penalties may prove beneficial, such as the parabolic function employed by Falcão and Borges [34, 38], and it may be that dynamic penalty coefficients should only be applied to certain constraints.

Future work could also evaluate the interaction between termination criteria and selection strategies. We found that instead using sequences of consecutive generations without improvement as the criterion would have led to a longer search (beyond 1500, 3000, and 6000, for landscapes A, B, and C, resp.) and would have likely yielded improvements, at least for proportional selection. With high-competitor tournament selection to the contrary, higher pressure to converge may trigger the criterion earlier, resulting in even worse relative performance.

Future work could also extend the model formulation to include additional planning concerns, such as road building and volume flows over time. Even-flow constraints can increase problem complexity when using exact techniques and the cluster-packing formulation [5], but in the author's collective experience even-flow constraints are not the most limiting or difficult constraints to satisfy when using metaheuristic solution techniques. In related applications, authors employing GA have not reported significant difficulty in achieving volume-based constraints (e.g., [34, 36, 38, 42]). Further, in some harvest scheduling applications with ecological objectives authors have likewise excluded volume constraints (e.g., [31, 33]). We do not anticipate that the absence of such constraints from experimental results presented here would affect the major implications of this research.

As landscapes grew in size and number of units it became increasingly difficult to reach feasible solutions, especially with respect to the old-growth constraint. The degree to which this is influenced by initial age-class distribution is unknown, though for future work we could experiment with varying the initial age-class distribution. Our landscape simulation code did not intentionally create clumps of similarly aged harvest units, so it may be that from the outset the landscapes had very few possibilities for feasible patches through time. Because the algorithm using penalties alone had difficulty finding feasible solutions for larger problems, we opted to retain an archive of feasible solutions. The archive works just like an elitist archive, but instead stores known feasible solutions for possible reintroduction into the population at some future generation when the best solution is infeasible. The use of both elitist and feasible archives necessitates a replacement strategy, and future work could also investigate alternate strategies such as random replacement or replacement by age rather than by the worst fitness.

We make no claim that genetic algorithms or particular selection/penalty schemes are superior for this or any other class of forest planning problem, but do recommend that practitioners consider more than just the canonical version when using genetic algorithms. We also recommend that the field of forest planning integrate with evolutionary computation advancements in order to stay on cutting edge of heuristics (as we should stay on the cutting edge of exact methods). For many practitioners solving complex spatial problems, use of exact solutions is simply not an option. Sessions et al. [67], for instance, present a spatial planning model for the Tillamook State Forest in Oregon that includes over a million integer decision variables. At a time where planning is increasingly incorporating spatially oriented ecological goals, there is a need within the forest planning community to continue development of appropriate, proven heuristics. The wealth of literature from nonforestry disciplines devoted to genetic algorithms and the broader field of evolutionary computation is a valuable resource that can inform forest researchers and planners seeking to design efficient and effective genetic algorithms to facilitate planning. The wide variety of possible implementations highlights the "art" of heuristic design, emphasizing the need for initial exploration and experimentation and thoughtful design.

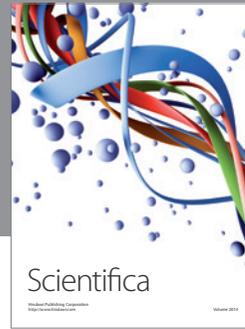
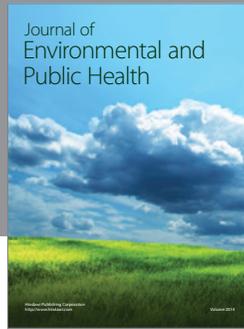
## Acknowledgments

The authors are grateful to the associate editor and three anonymous reviewers for their thoughtful and insightful comments that without doubt improved the quality of this manuscript. The primary author completed this work while a graduate student at Oregon State University as the recipient of the Wes and Nancy Lematta and the Dorothy M. Hoener fellowships. Preliminary results were presented at the Institute for Operations Research and Management Sciences (INFORMS) Annual Meeting, November 2007, Seattle, Wash, USA.

## References

- [1] P. Bettinger and W. Chung, "The key literature of, and trends in, forest-level management planning in North America, 1950–2001," *International Forestry Review*, vol. 6, no. 1, pp. 40–50, 2004.
- [2] S. F. Tóth and M. E. McDill, "Promoting large, compact mature forest patches in harvest scheduling models," *Environmental Modeling and Assessment*, vol. 13, no. 1, pp. 1–15, 2008.
- [3] S. F. Tóth, M. E. McDill, and S. Rebas, "Finding the efficient frontier of a bi-criteria, spatially explicit, harvest scheduling model," *Forest Science*, vol. 52, no. 1, pp. 93–107, 2006.
- [4] E. I. Ducheyne, R. R. De Wulf, and B. De Baets, "A spatial approach to forest-management optimization: linking GIS and multiple objective genetic algorithms," *International Journal of Geographical Information Science*, vol. 20, no. 8, pp. 917–928, 2006.
- [5] J. P. Vielma, A. T. Murray, D. M. Ryan, and A. Weintraub, "Improving computational capabilities for addressing volume constraints in forest harvest scheduling problems," *European Journal of Operational Research*, vol. 176, no. 2, pp. 1246–1264, 2007.
- [6] E. I. Ducheyne, R. R. De Wulf, and B. De Baets, "Single versus multiple objective genetic algorithms for solving the even-flow forest management problem," *Forest Ecology and Management*, vol. 201, no. 2-3, pp. 259–273, 2004.
- [7] I. Martins, M. Constantino, and J. G. Borges, "A column generation approach for solving a non-temporal forest harvest model with spatial structure constraints," *European Journal of Operational Research*, vol. 161, no. 2, pp. 478–498, 2005.
- [8] F. Caro, M. Constantino, I. Martins, and A. Weintraub, "A 2-opt tabu search procedure for the multiperiod forest harvesting problem with adjacency, greenup, old growth, and even flow constraints," *Forest Science*, vol. 49, no. 5, pp. 738–751, 2003.
- [9] S. Rebas and M. E. McDill, "A mixed-integer formulation of the minimum patch size problem," *Forest Science*, vol. 49, no. 4, pp. 608–618, 2003.
- [10] K. Öhman, "Creating continuous areas of old forest in longterm forest planning," *Canadian Journal of Forest Research*, vol. 30, no. 11, pp. 1817–1823, 2000.
- [11] K. Boston and J. Sessions, "Development of a spatial harvest scheduling system to promote the conservation between indigenous and exotic forests," *International Forestry Review*, vol. 8, no. 3, pp. 297–306, 2006.
- [12] G. Liu, S. Han, X. Zhao, J. D. Nelson, H. Wang, and W. Wang, "Optimisation algorithms for spatially constrained forest planning," *Ecological Modelling*, vol. 194, no. 4, pp. 421–428, 2006.
- [13] K. Boston and P. Bettinger, "Combining tabu search and genetic algorithm heuristic techniques to solve spatial harvest scheduling problems," *Forest Science*, vol. 48, no. 1, pp. 35–46, 2002.
- [14] M. E. McDill, S. A. Rebas, and J. Braze, "Harvest scheduling with area-based adjacency constraints," *Forest Science*, vol. 48, no. 4, pp. 631–642, 2002.
- [15] T. M. Barrett and J. K. Gillies, "Even-aged restrictions with sub-graph adjacency," *Annals of Operations Research*, vol. 95, no. 1–4, pp. 159–175, 2000.
- [16] P. Bettinger, K. Boston, and J. Sessions, "Intensifying a heuristic forest harvest scheduling search procedure with 2-opt decision choices," *Canadian Journal of Forest Research*, vol. 29, no. 11, pp. 1784–1792, 1999.
- [17] A. T. Murray, "Spatial restrictions in harvest scheduling," *Forest Science*, vol. 45, no. 1, pp. 45–52, 1999.
- [18] K. Boston and P. Bettinger, "An analysis of Monte Carlo integer programming, simulated annealing, and tabu search heuristics for solving spatial harvest scheduling problems," *Forest Science*, vol. 45, no. 2, pp. 292–301, 1999.
- [19] H. M. Hoganson and J. G. Borges, "Using dynamic programming and overlapping subproblems to address adjacency in large harvest scheduling problems," *Forest Science*, vol. 44, no. 4, pp. 526–538, 1998.
- [20] A. T. Murray and R. L. Church, "Analyzing cliques for imposing adjacency restrictions in forest models," *Forest Science*, vol. 42, no. 2, pp. 166–175, 1996.
- [21] C. Lockwood and T. Moore, "Harvest scheduling with spatial constraints: a simulated annealing approach," *Canadian Journal of Forest Research*, vol. 23, no. 3, pp. 468–478, 1993.
- [22] K. A. Crowe and J. D. Nelson, "An evaluation of the simulated annealing algorithm for solving the area-restricted harvest-scheduling model against optimal benchmarks," *Canadian Journal of Forest Research*, vol. 35, no. 10, pp. 2500–2509, 2005.
- [23] K. Crowe and J. Nelson, "An indirect search algorithm for harvest-scheduling under adjacency constraints," *Forest Science*, vol. 49, no. 1, pp. 1–11, 2003.
- [24] M. Goycoolea, A. T. Murray, F. Barahona, R. Epstein, and A. Weintraub, "Harvest scheduling subject to maximum area restrictions: exploring exact approaches," *Operations Research*, vol. 53, no. 3, pp. 490–500, 2005.
- [25] A. T. Murray, M. Goycoolea, and A. Weintraub, "Incorporating average and maximum area restrictions in harvest scheduling models," *Canadian Journal of Forest Research*, vol. 34, no. 2, pp. 456–464, 2004.
- [26] A. Weintraub and A. T. Murray, "Review of combinatorial problems induced by spatial forest harvesting planning," *Discrete Applied Mathematics*, vol. 154, no. 5, pp. 867–879, 2006.
- [27] J. Zhu and P. Bettinger, "Estimating the effects of adjacency and green-up constraints on landowners of different sizes and spatial arrangements located in the southeastern U.S.," *Forest Policy and Economics*, vol. 10, no. 5, pp. 295–302, 2008.
- [28] K. Boston and P. Bettinger, "An economic and landscape evaluation of the green-up rules for California, Oregon, and Washington (USA)," *Forest Policy and Economics*, vol. 8, no. 3, pp. 251–266, 2006.
- [29] F. Caro, R. Andalaft, X. Silva, A. Weintraub, P. Sapunar, and M. Cabello, "Evaluating the economic cost of environmental measures in plantation harvesting through the use of mathematical models," *Production and Operations Management*, vol. 12, no. 3, pp. 290–306, 2003.
- [30] K. Boston and P. Bettinger, "The economic impact of green-up constraints in the southeastern United States," *Forest Ecology and Management*, vol. 145, no. 3, pp. 191–202, 2001.
- [31] T. Pukkala and M. Kurttila, "Examining the performance of six heuristic optimisation techniques in different forest planning problems," *Silva Fennica*, vol. 39, no. 1, pp. 67–80, 2005.
- [32] H. D. Venema, P. H. Calamai, and P. Fieguth, "Forest structure optimization using evolutionary programming and landscape ecology metrics," *European Journal of Operational Research*, vol. 164, no. 2, pp. 423–439, 2005.
- [33] M. Palahí, T. Pukkala, L. Pascual, and A. Trasobares, "Examining alternative landscape metrics in ecological forest planning: a case for capercaillie in Catalonia," *Investigación Agraria: Sistemas y Recursos Forestales*, vol. 13, no. 3, pp. 527–538, 2004.

- [34] A. O. Falcão and J. Borges, "Combining random and systematic search heuristic procedures for solving spatially constrained forest management scheduling models," *Forest Science*, vol. 48, no. 3, pp. 608–621, 2002.
- [35] F. Lu and L. O. Eriksson, "Formation of harvest units with genetic algorithms," *Forest Ecology and Management*, vol. 130, no. 1–3, pp. 57–67, 2000.
- [36] D. S. Mullen and R. M. Butler, "The design of a genetic algorithm based spatially constrained timber harvest scheduling model," in *Proceedings of the 7th Symposium on Systems Analysis in Forest Resources*, pp. 57–65, USDA Forest Service, North Central Experiment Station, Traverse City, Mich, USA, May 1997, General Technical Report NC-205.
- [37] D. A. Hughell and J. P. Roise, "Simulated adaptive management for timber and wildlife under uncertainty," in *Proceedings of the 7th Symposium on Systems Analysis in Forest Resources*, Traverse City, Mich, USA, May 1997.
- [38] A. O. Falcão and J. G. Borges, "Designing an evolution program for solving integer forest management scheduling models: an application in Portugal," *Forest Science*, vol. 47, no. 2, pp. 158–168, 2001.
- [39] J. Hamann, *Optimizing the primary forest products supply chain: a multi-objective heuristic approach*, Ph.D. thesis, Oregon State University, Corvallis, Ore, USA, 2008.
- [40] A.-H. Mathey, E. Krcmar, D. Tait, I. Vertinsky, and J. Innes, "Forest planning using co-evolutionary cellular automata," *Forest Ecology and Management*, vol. 239, no. 1–3, pp. 45–56, 2007.
- [41] T. Heinonen and T. Pukkala, "A comparison of one- and two-compartment neighborhoods in heuristic search with spatial forest management goals," *Silva Fennica*, vol. 38, no. 3, pp. 319–332, 2004.
- [42] P. Bettinger, D. Graetz, K. Boston, J. Sessions, and W. Chung, "Eight heuristic planning techniques applied to three increasingly difficult wildlife planning problems," *Silva Fennica*, vol. 36, no. 2, pp. 561–584, 2002.
- [43] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, Mich, USA, 1975.
- [44] C. R. Reeves, "Genetic algorithms," in *Modern Heuristic Techniques for Combinatorial Problems*, C. R. Reeves, Ed., pp. 151–188, John Wiley & Sons, New York, NY, USA, 1993.
- [45] D. Dumitrescu, B. Lazzarini, L. C. Jain, and A. Dumitrescu, *Evolutionary Computation*, CRC Press LLC, Boca Raton, Fla, USA, 2000.
- [46] S. Legg, M. Hutter, and A. Kumar, "Tournament versus fitness uniform selection," Tech. Rep. IDSIA-04-04, Dalle Molle Institute for Artificial Intelligence, Manno, Switzerland, 2004.
- [47] A. Hertz and M. Widmer, "Guidelines for the use of meta-heuristics in combinatorial optimization," *European Journal of Operational Research*, vol. 151, no. 2, pp. 247–252, 2003.
- [48] E. W. Richards and E. A. Gunn, "Tabu search design for difficult forest management optimization problems," *Canadian Journal of Forest Research*, vol. 33, no. 6, pp. 1126–1133, 2003.
- [49] E. W. Richards and E. A. Gunn, "A model and tabu search method to optimize stand harvest and road construction schedules," *Forest Science*, vol. 46, no. 2, pp. 188–203, 2000.
- [50] S. Brumelle, D. Granot, M. Halme, and I. Vertinsky, "A tabu search algorithm for finding good forest harvest schedules satisfying green-up constraints," *European Journal of Operational Research*, vol. 106, no. 2–3, pp. 408–424, 1998.
- [51] FSC, "Forest Stewardship Council: Standards and Criteria," 2008, [http://www.fscus.org/standards\\_criteria](http://www.fscus.org/standards_criteria).
- [52] A. Osyczka, *Evolutionary Algorithms for Single and Multicriteria Design Optimization*, Physica, Heidelberg, Germany, 2002.
- [53] T. Bäck, D. B. Fogel, and T. Michalewicz, *Evolutionary Computation I: Basic Algorithms and Operators*, Institute of Physics, Bristol, UK, 2000.
- [54] D. E. Goldberg and K. Deb, "A comparative analysis of selection schemes used in genetic algorithms," in *Foundations of Genetic Algorithms*, G. J. E. Rawlins, Ed., pp. 69–93, Morgan Kaufmann, San Mateo, Calif, USA, 1991.
- [55] G. B. Fogel and D. B. Fogel, "Continuous evolutionary programming: analysis and experiments," *Cybernetics and Systems*, vol. 26, no. 1, pp. 79–90, 1995.
- [56] J. Dewey, F. Maier, W. D. Potter, et al., "Prescriptive treatment optimization using a genetic algorithm: a tool for forest management," in *Proceedings of the 3rd Biennial Meeting of the International Environmental Modelling and Software Society (IEMSs '06)*, A. Voinov, A. Jakeman, and A. Rizzoli, Eds., pp. 1–6, International Environmental Modelling and Software Society, Burlington, Vt, USA, July 2006.
- [57] L. S. Davis, K. N. Johnson, P. S. Bettinger, and T. E. Howard, *Forest Management: To Sustain Ecological, Economic, and Social Values*, McGraw Hill, New York, NY, USA, 2001.
- [58] R. Battiti and G. Tecchiolli, "The reactive tabu search," *ORSA Journal of Computing*, vol. 6, no. 2, pp. 126–140, 1994.
- [59] M. Gendreau, A. Hertz, and G. Laporte, "A tabu search heuristic for the vehicle routing problem," *Management Science*, vol. 40, no. 10, pp. 1276–1290, 1994.
- [60] F. Glover, "Tabu search: a tutorial," *Interfaces*, vol. 20, no. 4, pp. 74–94, 1990.
- [61] R. Turner, "deldir: Delaunay Triangulation and Dirichlet (Voronoi) Tessellation," R package version 0.0-7, 2007, <http://www.R-project.org>.
- [62] R Development Core Team, "R: a language and environment for statistical computing," R Foundation for Statistical Computing, Vienna, Austria, 2008, <http://www.R-project.org>.
- [63] A. Okabe, B. Boots, K. Sugihara, and S. Chiu, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, John Wiley & Sons, West Sussex, UK, 2000.
- [64] A. R. Stage, "Prognosis model for stand development," USDA Forest Service Research Paper INT-137, Intermountain Forest and Range Experiment Station, Ogden, Utah, USA, 1973.
- [65] P. Bettinger, J. Sessions, and K. Boston, "A review of the status and use of validation procedures for heuristics used in forest planning," *International Journal of Mathematical and Computational Forestry & Natural-Resource Sciences*, vol. 1, no. 1, pp. 25–36, 2009.
- [66] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.
- [67] J. Sessions, P. Overhulser, P. Bettinger, and D. Johnson, "Linking multiple tools: an American case," in *Computer Applications in Sustainable Forest Management: Including Perspectives on Collaboration and Integration*, G. Shao and K. Reynolds, Eds., vol. 11 of *Managing Forest Ecosystems*, chapter 11, pp. 223–238, Springer, New York, NY, USA, 2006.



**Hindawi**

Submit your manuscripts at  
<http://www.hindawi.com>

