**A New Heuristic Method for Solving Spatially Constrained Forest Planning Problems Based on Mitigation of Infeasibilities Radiating Outward from a Forced Choice**

Anumita Srivastava

Computer Science

The University of Georgia

**Abstract:**

In this experiment, our aim is to schedule forest harvesting over time and space in such a way that the openings are restricted to a certain size and also aren’t adjacent to each other in the same time frame. This is done using a simple Genetic Algorithm.

**Introduction:**

A number mathematical programming techniques have been used to tackle the spatial forest planning problems. Other heuristic approaches include the sequential approach, the sequential quenching and tempering approach and hybrid heuristic approaches.

All of the search processes in general, can be grouped by the type of change that the forest plan goes through in each iteration and the way the units and their statuses are handled. In this experiment we will use a simple Genetic Algorithm to tackle the forest planning problem. A genetic Algorithm uses the techniques of evolution that are observed in nature to get the best solutions to solve a variety of optimization and search problems. It uses operators such as selection, crossover and mutation to explore a search space and to exploit the solution that is finds the best. All potential infeasibilities are subjected to penalties.

**Forest Planning Problem**

The objective function for the forest planning problem is represented as:



where *t* = a time period

*Ht* = the total harvest volume (tons) during a time T

period *T* = a target harvest volume

This function finds the solution that has the highest harvest volume for all three time periods namely H1, H2 and H3.

**Experiment Setup**

We get the input from two files.

1. The volume file: This tells us the harvest that is reaped for every stand in different time periods H1, H2 and H3. For example, for stand 1, the file has the following entry:

1,5.69,29,43,55

This means that stand 1, covers 5.69 acres and produces 29,43 and 55 per acre harvest volume in their respective harvest periods.

There are a total of 73 such stands.

1. Adjacency file: This tells us about a stand’s neighborhood. For example, 1,6;1,34;1,48 entries from this file means that stand 1 has stands 6,34 and 48 adjacent to itself.

The volume file is used to calculate the fitness whereas the adjacency file is used to form the adjacency matrix. By maintaining adjacency matrix, we ensure that no two adjacent stands are cut in the same time period.

We use T = 34467 for our fitness calculation.

**Representation:**

We use the 219 representation for representation.

1. Each stand has a certain likelihood associated with every Harvest period. This means that there is a random number generated between 0 to 1 and this number represents the likelihood of that particular stand being cut in that harvest period.

For example, for time period 1, the likelihood of it getting cut in H1 is 0.2, the likelihood of it getting cut in H2 is 0.1 and the likelihood of it getting cut in H3 is 0.8.

This would mean that the likelihood of stand 1 getting cut is maximum in harvest period 3.

1. The next step is to sort the likelihoods from the highest to lowest such that each stand would get cut in the harvest period that has the greatest value of likelihood associated with it.
2. We then get a schedule that has 73 harvest period values for each of the stands.

An important step is to introduce the NO CUT mechanism. In cases where adjacent stands are placed in the same time period, we make the time period 0 in the schedule, thereby implying that that particular stand cannot get cut in that harvest period.

An example of this schedule can be, 1,2,0,3,3,0,0,1,3,2,0,2,0,1,0,0,3,3,0,2,1,1,1,1,3,3,1,2,1,0,3,3,2,0,0,2,0,0,2,3,3,3,3,3,1,1,1,3,0,2,1,2,0,1,3,2,0,2,2,3,0,0,0,1,2,2,2,1,0,0,0,0,1,2,3,1,1.

Each integer in the schedule tells us the harvest period in which the respective stand will be cut.

The following parameters were used in the Genetic Algorithm:

1. Population size = 5000 – 20,000
2. Crossover rate = 0.6-0.8
3. Mutation rate = 0.01 – 0.03
4. Elitism = False
5. Selection Method = Tournament Selection
6. Stopping Criteria = 20 Stable generations

**Results:**

We ran the Genetic algorithm until 20 stable generations with varying population sizes from 5000-20,000.

The crossover rate is kept at 0.8 and mutation rate at 0.01 to find the best schedule and an optimal value of the fitness function.

|  |  |  |
| --- | --- | --- |
| Population size | Best Fitness Value | Schedule |
| 5000 | 1.01855996262838E8 | 212030320130031020103212331212203003000  1300201330302030032031020030230000 |
| 10,000 | 1.51695371328722E8 | 101032210300301003301232020130032002210  2110013100001200020101012300320312 |
| 15,000 | 7.78828730121791E7 | 3030311322301312201000013303312002233322010  001301100200321012002123002000 |
| 20,000 | 1.59547798106354E7 | 13203200110033021131003210012222000101320  00200001200300110321030200030102 |

We get the best forest schedule for a population size of 20,000 after 2 million fitness evaluations.

**Conclusion:**

We implemented the Forest Planning Problem using a Simple Genetic Algorithm. We use a 219 representation where we assign random likelihoods to each stand getting cut in a particular harvest period. We introduced a “no cut” option so as to adhere to the adjacency constraints.

**PSO:**

**Experiment Setup:**

For the next part of our experiment, we use Particle Swarm Optimization to generate a schedule of forest harvesting over time and space.

We use the following parameters for our experiment:

Swarm Size = 50

Number of generations = 150

The velocities lie in the range [-4,4].

The cognitive and social influences are fixed at 2.

The location is denoted by x, y and lie in the range [-6,6].

The inertial is fixed at 1.2

**Results:**

With the above stated PSO parameters, we get the best fitness value as 1.392235815245313E7 with the corresponding schedule as:

2310101230300021010303232203020203030300311333210022020132020103302113000