Advanced Forest Planning

Dr. Bettinger

Chaz Merritt

09/28/2024

Assignment 4 – Genetic Algorithm

I decided to implement a genetic algorithm to solve the Hypothetica optimization problem. This is a population-based method that works extremely quickly and effectively. Like many others, it is inspired by nature. The hardest part was dealing with constraint violations, as Genetic Algorithms are stochastic due to their mutation and crossover functions. To address this, mutations were done as swaps only, to largely preserve feasibility while still introducing diversity. Similarly, crossover was done as a one-point crossover. The most important however, was to initialize the population using a greedy volume-based approach, similar to binary search. The fitness function calculated the squared deviations from the target, as well as applied a large penalty to harvest schedules that had constraint violations. In this way, the genetic algorithm is allowed to somewhat explore unfeasible solutions, though in my experimentation, always converged to a highly optimal and feasible solution. Selection was implemented by using tournament selection for parents. Elitism was also implemented so that the best individuals would carry over to the new populations across generations. I was also able to continue developing and using my HarvestSchedule class, and it is now in an even better position for reusability than before. This was challenging to implement overall, but now that it is done, I believe it is my strongest work so far for this course. The deviations are negligible even with small population sizes and small number of generations.

I found that the algorithm initializes a population that is extremely close to the optimal in the very beginning and doesn't stray away from these strong initial solutions over the course of 100 generations or more. This is likely due to the greedy volume-based initialization method which randomly adds stands to the period until the target is reached, all the while checking for constraint violations.

Below is the output of successive runs of the program.

Population Size = 100, Generations = 100

Volume Harvested by Period:
Period 1: 11896.00 cubic feet
Period 2: 11860.00 cubic feet
Period 3: 11792.00 cubic feet
Period 4: 12080.00 cubic feet
Period 5: 11976.00 cubic feet
Period 6: 12240.00 cubic feet
Period 7: 11772.00 cubic feet
Period 8: 12124.00 cubic feet

Harvest Schedule:
Period 1: Stands 108, 38, 112, 106, 13, 51, 94, 91, 21, 42, 59, 73, 80, 111
Period 2: Stands 103, 43, 37, 47, 50, 3, 27, 32, 110, 48, 24
Period 3: Stands 1, 54, 75, 99, 71, 66, 86, 55, 30, 18, 105
Period 4: Stands 44, 41, 83, 36, 74, 26, 93, 9, 12
Period 5: Stands 76, 107, 56, 67, 92, 101, 7, 34, 39, 61, 4
Period 6: Stands 68, 85, 45, 46, 60, 49, 70, 15, 16, 77, 19
Period 7: Stands 72, 31, 65, 57, 69, 104, 22, 109, 79, 82, 6
Period 8: Stands 64, 8, 88, 33, 23, 81, 10

Population size = 1000, Generations = 500
Volume Harvested by Period:
Period 1: 11852.00 cubic feet
Period 2: 11996.00 cubic feet
Period 3: 11796.00 cubic feet
Period 4: 11888.00 cubic feet
Period 5: 11772.00 cubic feet
Period 6: 11732.00 cubic feet
Period 7: 12044.00 cubic feet
Period 8: 11948.00 cubic feet

Harvest Schedule:
Period 1: Stands 88, 99, 66, 9, 43, 55, 51, 73, 111, 13, 49, 61, 37, 42, 53, 12, 110
Period 2: Stands 32, 47, 36, 94, 75, 103, 45, 59, 80, 68, 38
Period 3: Stands 56, 44, 91, 108, 50, 3, 86, 30, 105, 24, 102, 98
Period 4: Stands 70, 72, 67, 1, 23, 77, 74, 41, 64, 95, 21
Period 5: Stands 93, 106, 31, 60, 27, 92, 89, 4, 54, 39, 7, 5, 26, 19, 90, 63
Period 6: Stands 18, 78, 28, 112, 16, 104, 57, 15, 52, 25, 85, 96, 100, 82
Period 7: Stands 35, 113, 84, 107, 40, 69, 83
Period 8: Stands 79, 71, 76, 33, 101, 87, 2, 81