Nathan Fox

Artificial Intelligence

Due Monday 10/7

Genetic Algorithms Homework

Experiment

I ran the GA.m program a number of times to evaluate the effects of three different variables: path length, number of genes in pool, and number of generations. My default values were worldsize\*2, 50, and 60 (respectively). I ran each test twice (not enough for a good sample, but I didn’t have a lot of time) and recorded the average of the runtimes and solution costs.

Results

|  |  |  |
| --- | --- | --- |
|  | Runtime | Best Solution |
| Path\_Len |  |  |
| 1 | 18.55 | 23.33 |
| 2 | 18.783 | 57.58 |
| 3 | 19.246 | 119.92 |
| Num\_Genes |  |  |
| 10 | 17.7 | 59.09 |
| 50 | 18.588 | 57.31 |
| 100 | 19.83 | 53.13 |
| 500 | 30.098 | 48.63 |
| Max\_Generations |  |  |
| 30 | 9.3 | 67.03 |
| 60 | 18.511 | 57.21 |
| 120 | 36.862 | 51.58 |
|  |  |  |
| (1, 500, 200) | 78.8 | 21.41 |
| (2, 250, 100) | 39.458 | 53.39 |

Conclusions

1. Modifications to path length

Changing path length had little effect on runtime. This makes sense because the program is still processing the same size of gene pool and the same number of generations; the genes are just bigger. However, if determining a gene’s fitness required polynomial time (based on the length of the gene) or worse, then changing the path length would have a greater effect on runtime.

Changing path length had a great effect on the best solution found. This makes sense because a path twice as long should cost at least twice as much. However, the data shows that when path length is multiplied by *n*, the best solution multiplies by more than *n*. This is because making the path longer (even by just a little) adds a lot more possible solutions, thus making it less likely for the algorithm to find one of the best solutions.

2. Modifications to size of gene pool

Changing the number of genes in the gene pool does affect the runtime. Each gene’s fitness has to be evaluated, which takes time. If determining fitness were computationally difficult, the effect on runtime would be even greater.

Changing the number of genes did tend to affect the best solution as well. As the gene pool grows, the probability of finding one of the best genes increases.

3. Modifications to the number of generations

Changing the number of generations has a huge effect on runtime. This makes sense because the number of generations is the number of iterations of the algorithm, so twice as many iterations should take twice as much time, like the data shows.

Changing the number of generations also affects the best solution. Adding generations adds more mutations and crossovers of the top genes into the gene pool, thus increasing the probability of finding one of the best genes.