CS 5722/CEE 5290/ORIE 5340

Heuristic Methods for Optimization

Homework 12: Genetic programming and Response Surface Optimization

Assigned:Wednesday, November 23, 2011

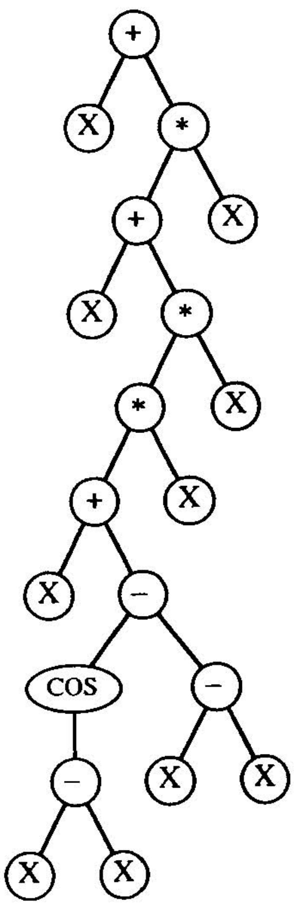
Due: Friday, December 2, 2011 (can be extended to Dec. 5 without penalty)

TA office hours: Tue, Thu 2:00 – 3:30

Professor Shoemaker's office hour: Mon, Wed 2:30 – 3:30

(Not all of these problems will be graded. All of these topics could be on the final.)

1. **Symbolic Regression**



The parse tree on the left represents the best solution from Generation 34 to a symbolic regression for

$$f(x) = x^4 + x^3 + x^2 + x$$

(Slide 30 from the 11-1 handout)

(a) Show that the parse tree represents f(x) exactly by writing out the exact mathematical equation represented by the parse tree and then combining terms so that you finally get

$$f(x) = x^4 + x^3 + x^2 + x$$

(b) Are there unnecessary terms in the parse tree?

2. **USING GENETIC PROGRAMMING SOFTWARE**

This problem set may be done with GPLAB - A Genetic Programming Toolbox for MATLAB (by Sara Silva) which is freely downloadable from http://gplab.sourceforge.net/. The data files can be downloaded from blackboard. [Hint: make changes in the 'demo.m' file in the toolbox which is the solution of a different symbolic regression problem. Also, the manual is really helpful !]

SYMBOLIC REGRESSION OF $X^2/2 + 2X + 2$

Use a function set consisting of the addition (+), subtraction (-), multiplication (*), and protected division (%) to do a symbolic regression of the target function $X^2/2 + 2X + 2$. The protected division function % is protected against division by zero. Do not include random constants in the terminal set. Use a population of size of M = 1,000. Use Maximum number of generations to be run, G, of 151. Set:
crossover_fraction = 0.90,
copy_fraction = 0.10, and
mutation_fraction = 0.00.

QUESTIONS:
(a) Symbolic regression of $X^2/2 + 2X + 2$:**.**Fill in the following tableau for this problem.
Tableau for symbolic regression of $\mathbf{X^2/2 + 2X + 2}$

Objective:
Terminal set
Function set:
Fitness cases:
Raw fitness:
Standardized fitness:
Hits:
Wrapper:
Parameters
Success predicate

(b) Generation 0 performance: What is the best-of-generation individual of generation 0s fitness value, depth of tree and number of nodes?

(c) Pick out some interesting-looking and explainable best-of-generation individual for some intermediate generation that illustrates progress toward the solution. Write out the individual, its generation number, and its fitness. Why this individual has an intermediate value of fitness (i.e., what is there about its structure that gives it intermediate fitness, as opposed to best, as opposed to what you saw at generation 0)?

(d) The Best-of-Run Individual: Write the generation number of the final generation of the run, the best-of-generation individual of this final generation, and its fitness value.

(e)Was this a perfect (i.e., algebraically correct) or approximate solution to the problem?

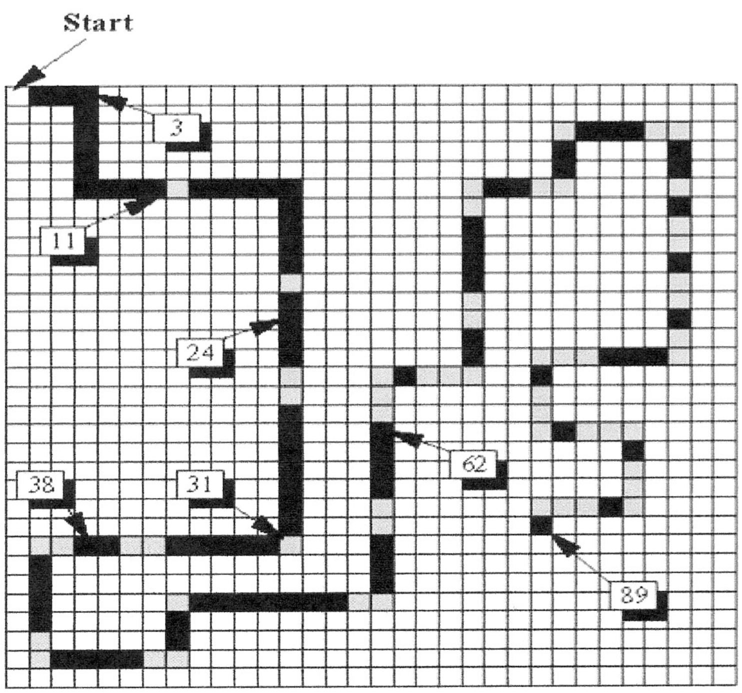
(f) Try running this problem at least more time (or until one run yields an algebraically correct solution). Write the generation number of the final generation of the run, the best-of-generation individual of this final generation, and its fitness value.

3. **Genetic programming:** In the lectures on genetic programming the definitions of functions, terminals, parse trees, and the Lisp function *ProgNm* (for arbitrary integer m) were given. Below is shown the Sante Fe Trail and an alternate trail, the Los Altos Hills

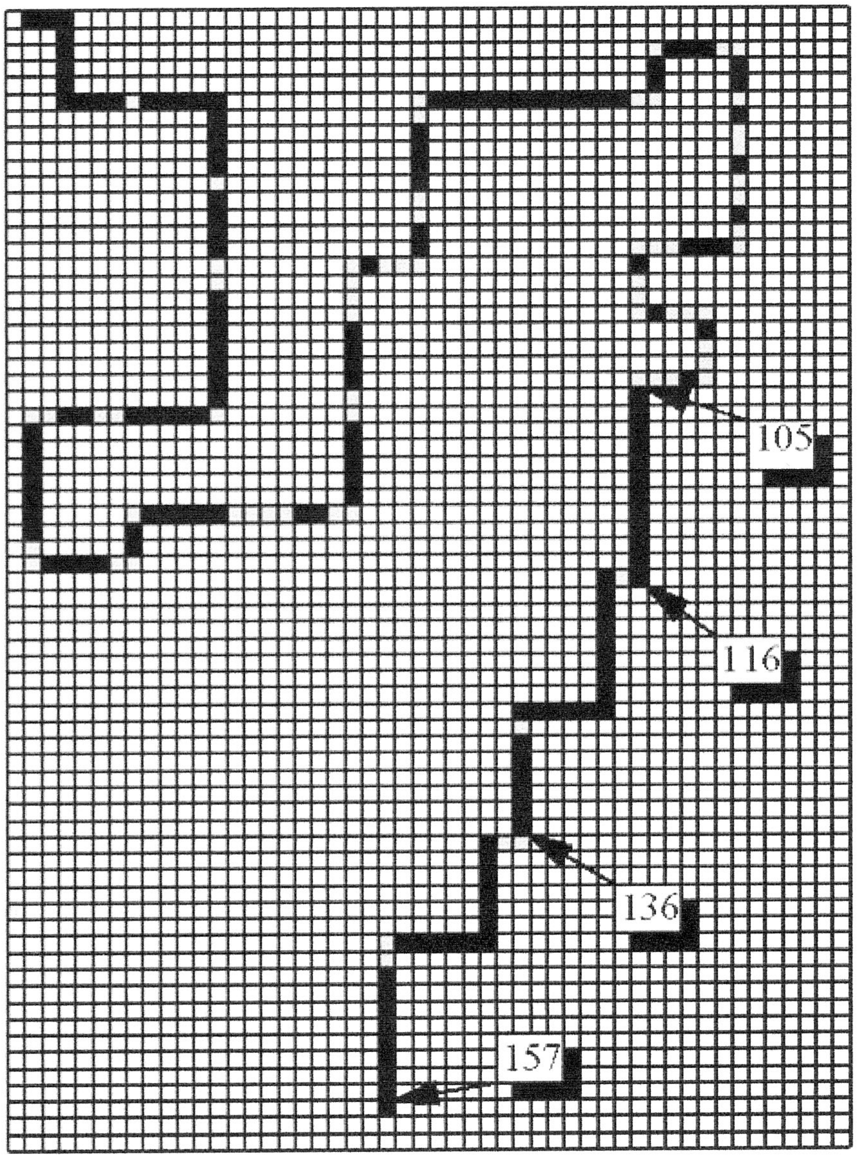
trail is on the next page. There are situations at locations 116 and 136 on the Los Altos trail that do not occur on the Sante Fe trail.

a. Explain why the optimal policy for the Sante Fe Trail will not be optimal for the Los Altos trail.

b. Assume you want to structure a genetic programming search for the optimal solution for the Los Altos trail. For the Los Altos trail, there is more food available (157 units), more grid cells, and you allow more time for a search for food (3000 time steps). Explain what your functions and terminals would be in a Genetic Programming search and how this would differ from the functions and terminals used for the search for the Santa Fe trail. Explain the reason for your answer. You do not need to provide the actual parse trees!



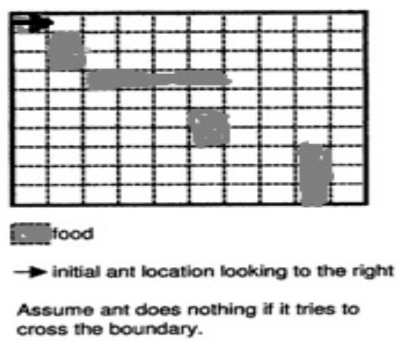
The Santa Fe Trail for the artificial ant problem.



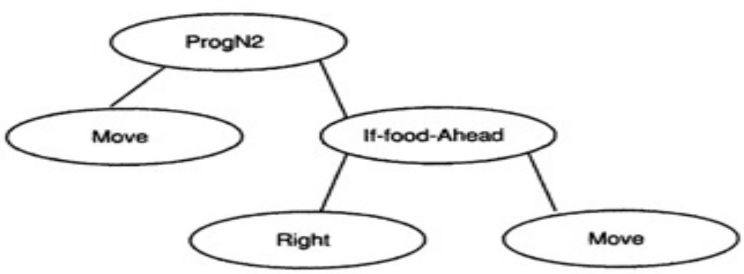
The Los Altos Hills trail for the artificial ant problem.

4. **Genetic Programming:** Assume you had an ant trail that you know has two corners that turn left and no corners that turn right. Also you know have food gaps of one, two , three or four adjacent spaces and these only occur on the straight part of the path. In other words there are <u>no gaps in food at the corners or within one space before or after the corner</u>. Give the parse tree that you think would be able to go through this path quickly with no unnecessary circles in the parse tree. (You are supposed to use your own human intelligence to solve this problem and do not need to implement a computer code. But of course you might get some ideas from the parse trees you have seen for other trails.)

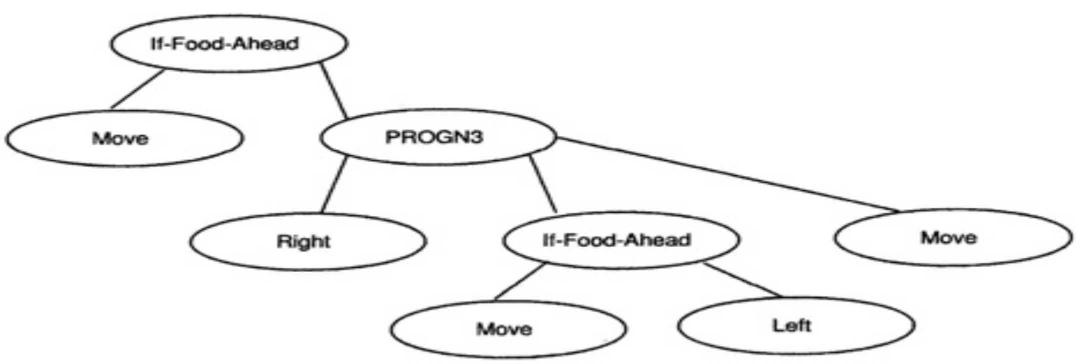
5. For the food diagram below, count how many pieces of food are eaten by an ant following solutions a) and b) (represented as parse trees) after 10 time steps? One time step passes when one of the terminal set commands is executed.



a)



b)



6. **Response surface methods (to be discussed starting Nov. 28).**

   Figure 4 in the lecture handout describes results for 6 algorithms. Briefly describe what each of these algorithm assumes in terms of

   a) Number of children generated
   b) Number of children for which expensive function evaluation is done
   c) Whether a response surface is used?

   What do the results in the figure tell you? Is it computationally advantageous to use surrogate response surfaces coupled to a heuristic? Explain based on the information provided.

Reference for Problem 1 (which you are not required to read): Regis, R., C.A.Shoemaker, "Local Function Approximation in Evolutionary Algorithms for the Optimization of Costly Functions**,"** ***IEEE Transactions on Evolutionary Computation*** 8 (5) 490-505, 2004