Lund University
Dept. of Computer Science

EDA 132/DATE 15
VT 2009

Assignment 4C: Genetic Algorithms

Goal To get acquainted with the idea and implementation of genetic algorithms. To get a feeling of how the population size and the degree of mutation will affect the result.

Reading advice Russell & Norvig: Artificial Intelligence - A Modern Approach, Section 4.3 in the 2nd edition.

Deadline The result must be handed in for evaluation before 23.59 Thursday, April 30, 2009.

Your task

Implement a program in the programming language of your choice to solve task 1 and task 2. With careful design choices, the same core program can be used for both tasks.

After implementing the program, test the tasks using different combinations of population sizes and mutation probabilities. Summarize these results in a table and discuss the implications. Remember to use a proper measure to compare the results, e.g., the total number of patterns tested rather than the number of generations, or simply the execution time used.

Results to be handed in

1. A brief description of your implementation including comments on your design choices and how your selection, mutation and cross-over works.
2. The result, i.e., a table summarizing the combinations of mutation probabilities and population sizes, and your reflections on the results. Please also tell which pattern that was finally selected and give its fitness and content (and weight in task 2).
3. The path to where your program is located and instructions on how to run it.

Grading

In order to pass this assignment, both tasks must be done well. The grading is based primarily on the quality of the report handed in, how well you manage to use your program as a tool, how well you identify reasonable tests to make, and how well you manage to make a clever analysis of the results.

Note

Make sure that your executable program as well as all the directories in the path are accessible to world (`chmod 705 <filename>`).

Task 1 - A trivial 10-variable minimizing problem

Mimimize the function:

$$f(x_1, \ldots, x_{10}) = \left(\sum_i x_i^2 \right),$$

where $-5.12 \leq x_1, \ldots, x_{10} \leq 5.12$.

Fitness

In a minimizing problem, fitness is reversed compare to a maximizing problem. You can easily negate the function f itself to use as fitness.

Selection

You may use the standard selection method where the probability of selection is proportional to the fitness. Usually, a selected pattern is not removed from the selection set (i.e., the old generation) but is allowed to take part in the subsequent selections as well, i.e., a pattern with a high fitness value has a high probability to be selected several times.

Cross-over

Two patterns are selected as parents to produce one or two new offsprings by means of cross-over. One or several cross-over points are randomly chosen. Cross-over points should, depending on your representation, often be made between two values, e.g., x_i and x_{i+1}. The process of cross-over may be repeated for several pairs of patterns.

Mutation

To mutate a pattern, one of (or any mumber of) the values are replaced by a new value within the allowed interval. For a smoother operation, the new value should not be selected from the entire interval but from a smaller intervall around the original value.

Task 2 - A knapsack problem

Picture yourself as a thief surrounded by precious treasures. You would like to take each and every one of them. Unlucky for you, there is a limit on how much you can carry. So you are forced to leave some treasures behind. Now you have to decide which treasures to take and which treasures to leave. You want, of course, to maximize the value of the treasures you bring home.

Let w_i denote the weight for item i , p_i the value of item i , and x_i be 1 if you carry the item and 0 if you leave it behind. Further, let C denote the maximum load you are able to carry. This means that you need to maximize $\sum_i p_i x_i$ while keeping the constraint $\sum_i w_i x_i < C$.

For this task you should set $C = 400$ and use the 50 item weights and values given in the file `Values.java` that you will find on the course home page.

Special care

For this assignment you have to take special care when creating or modifying patterns by mutation or cross-over to ensure that the constraint is fulfilled at all times. To find out a way to accomplish this is part of the assignment.

Cross-over

Find a method by which one or two new patterns are based on two parents. You can use cross-over points but feel free to design other means of breeding. Remember to make sure that the offspring meets the constraints.