Fall 2018 COSC **3P71 Artificial Intelligence: Assignment 2**

Instructor: B. Ombuki-Berman

Tutorial leader: James Paterson

Teaching Assistants: Jocko Pawel & James Paterson

Available Date: Wednesday October 10th, 2018

Source Code Due: Friday, Nov. 2th, 3:00 pm, **NO LATES**.
Full-Report Due: Monday, Nov. 5th, 3:00 pm, **NO LATES**

Goal: Genetic algorithm implementation and application.

Languages: Any programming language of your choice.

Task: Implement a genetic algorithm system for the Traveling Salesman Problem described in class using the **two data sets** provided at:

The overall procedure for a simple GA will be something like this:

Procedure simple GA
begin
Read problem instance data;
set GA parameters;
generate randomly an initial population POP of size Pop_Size;
  for gen = 1 to MAXGEN **do**
  evaluate fitness of the individuals of POP;
  select new population using some selection strategy;
  apply genetic operators, crossover and mutation;
 endfor;
end;

The GA's main modules are described as:

(a) **Initial Population initializer:** creates a population of size Pop_Size of randomized individuals as described in class.

(b) **Reproduction**: use Tournament Selection (remember K = 2, 3, 4 or 5)

(c) **Crossover**: given two individuals, this creates two offspring. Implement your GA using the following crossover strategies independently:

(i) Uniform order crossover (UOX, with bit mask)
(ii) A second crossover of your choice. Of course, the crossover must ensure valid offspring chromosomes for the TSP problem. Some examples are: partially mapped crossover (PMX), Order crossover (OX), cycle crossover (CX), two-points crossover-with repair, any other etc.,

(d) **Mutator**: given an individual, this creates a mutated individual. Implement your GA using a mutation operator of your choice (from those discussed in class). Again, the mutation operator must ensure a valid chromosome for the TSP problem!

(e) **Fitness evaluation function**: Total distance traveled, and recall that the last city visited is connected to the first city (the shorter the distance, the more fit an individual is)

(f) **Genetic algorithm system**: This is the implementation of the GA system

(g) **user parameters:** population size, maximum generation span, probability of (crossover, mutation etc)

Your GA program should permit the user to easily define his/her own genetic parameters (e.g., crossover rate, mutation rate, population size, maximum generation span etc.….). The program should allow testing for various problem sizes, i.e., size of cities, n.

Experimental Analysis

1) Run your GA to compare the performance of the two crossover operators mentioned above by using the following parameters (and include elitism in all cases):
    a. Crossover rate = 100 %, mutation = 0%
    b. Crossover rate = 100 %, mutation = 10%
    c. Crossover rate = 90 %, mutation 0%
    d. Crossover rate = 90 %, mutation 10%
    e. Determine your own best parameter settings

    (PS. For elitism), first consider an elite strategy where only the best chromosome is replicated to the next generation. Next consider replication criteria where a certain percentage of individuals determined empirically (no greater than 10%) are allowed to replicate (include chromosome of best fitness value for this replication)

2) Incorporate into your experiments your own innovative☺ idea; this would be a different initial population representation & creation, a different selection scheme, a different (third) crossover not discussed in class, etc. Your idea could be introducing some local search into your GA etc (This will be for **bonus** mark of 2% of total course grade)

For each experiment mentioned above, run your GA at least 5 times. Use the data set "provided" obtained from TSP-online benchmark data at:

http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsp/ulysses22.tsp
http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsp/eil51.tsp

Output the following to a file or standard output:

a) All GA parameters, including random number seed
b) Per each generation: best fitness value, average population fitness value
c) Per each run: best solution fitness and its corresponding best solution chromosome

Finally compute for the multiple runs: average of best fitness per generation and average population fitness per generation. Using a graph drawing tool such as excel, plot well labeled graphs for experiment 1, above including experiment 2 (if done). Types of graphs you plot for experiment 2 depend on your incorporated idea, if any. Feel free to experiment with different crossover and mutation rates. You will set your own Pop-Size and generation size.

Lastly, prepare a summarized report **using IEEE format** introduced to you at the tutorial. The link for IEEE format details is found at:

https://www.ieee.org/conferences/publishing/templates.html

Use the sub-headings (a) Objective and problem definition b) Summary parameters used (c) Results: that is, summary tables and graphs, EXPLAIN your graphs in detail. (d) Detailed discussions and conclusions from your results. Your discussions should include issues like which crossover performed better than the other one, if more than one mutation type tried, which one performed better. If you included local search, did it help? How did the choice of GA parameters affect the final outcome etc? This report is very important, so be sure to include it. Start early, gathering the data and doing the experimental analysis will take much more time than coding the assignment.

NOTE: You are allowed to discuss the assignment with friends, but you must do the work independently