CS 4623/6623 **Evolutionary Computation** **Spring 2023**

Wainwright

Software Project

Part I

You are to independently program from “scratch” a **Simple Genetic Algorithm** (SGA) designed specifically for the problem you have chosen (or was assigned to you) that implements at least **two different selection techniques** (roulette, tournament, rank), at least **two different crossover** operators for your problem, and at least **two different mutation** operators for your problem. In addition in your SGA you must be able to initially specify (a) the crossover rate, (b) the mutation rate.

Considerations for Implementing A Simple Genetic Algorithm (SGA)

A. For your assigned problem you must first determine:
1. The representation for the chromosome (bit string, permutation of integers, etc.)
2. The fitness function
3. The population size
4. If elitism is implemented or not. **(Note: Elitism should always be implemented)**.

B. Decide how your GA will terminate
1. After a certain amount of time
2. After a certain number of generations
3. When the population has converged or variance of the fitness values for the chromosomes is within a specified amount, such as all within 1% of each other.
4. Some combination of the above three options, or **which ever comes first**

C. Ordinarily, you would need to decide if you are going to implement a Generational GA of a Steady-State GA. **(For this project you are to implement the Generational GA)**

D. Determine how you are going to generate the initial population (randomly or by some other means). If by some other means, you must justify why you think this is a good idea.

E. Implement at least two selection techniques (Roulette, Rank, Tournament, etc)

F. Crossover
1. Determine the crossover rate (usually 80% to 100%). 100% is common
2. Implement at least two different crossover techniques and use in your SGA.

G. Mutation
1. Determine the mutation rate (usually 1% to 5%)
   Mutation rate could be adaptive. (explained in class)
2. Implement at least two different mutation operations and use in your SGA

H. Datasets

You are responsible for finding or generating your own set of test data. There needs to be small, medium and multiple large problems sets to test your algorithms. Most of the problems listed below have example data sets from various web sites, for example:
http://www.rpi.edu/~mitchj/sites_or.html

This is also a good source of other Combinatorial Problems that you might consider choosing. A good source for location possible problem sets can be found at the **OR Library**:
http://people.brunel.ac.uk/~mastjjb/jeb/info.html

Even if you cannot find datasets from various web sites, it is easy to randomly generate your datasets. I can help you with this.

You must convince me that your SGA is working. You do this by contriving a small data set with a known optimal, then determine if your SGA can find the optimal solution. If so, you are OK to move on to medium and large datasets where the optimal may not be known. If your SGA cannot find the optimal solution for the contrived dataset, then you have a serious error(s) somewhere in your SGA. I will discuss contrived datasets in class and give several ideas on how to generate such datasets.

Part II

You are to independently program from "scratch" a **Simulated Annealing Algorithm** (SA) designed specifically for the problem you have chosen (or was assigned to you) that implements at least two different perturbation operators. The perturbation operators are most likely the same as the two mutation operators implemented for your SGA.

Part III

You are to independently program from scratch a **"Foolish" Hill Climbing Algorithm** designed specifically for the problem you have chosen (or was assigned to you). This is very easy to do. Use the Simulated Annealing Algorithm that you developed and remove the option of allowing for a worse solution. That is, this version of the general Hill Climbing Algorithm does not allow worse answers under any circumstances. This is a minor change to the SA algorithm for your problem. This change should produce inferior results; but things are random, so there is a small chance it may not.

Part IV

Turn in a written report. A highly suggested content and outline is provided below.

Part V

Each Graduate student will prepare a PowerPoint presentation of the results and conclusions of their project and present it to the class (5-10 Minutes).

Turn in the following to your instructor:

Email to me (rogerw@utulsa.edu) all of the source code, data sets used, and your report, etc..
I need an electronic copy of all of your "stuff". This includes:

- A computer listing of all three programs, SGA, SA, “Foolish” Hill Climbing.
- A listing of your best runs (output) from each of your programs. **DO NOT** include all generations!
- A short report (5-10 pages or so) describing your entire project. The report should be in Word format or PDF format.

Suggested outline for your paper (report)

1. Introduction
   Clearly describe the problem you are solving.

2. Chromosome and fitness
Clearly describe the chromosome representation and the fitness function(s) that you used. In addition if infeasible chromosomes are possible in your problem, clearly describe how you handled them. If there is an encoding scheme for your chromosome, describe that.

3. Operators
   Clearly describe the reproduction (selection) operators you used, the crossover operators you used and the mutation operators you used in your GA. Clearly describe the perturbation functions used in your SA.

4. Parameters used
   Clearly specify your GA parameters: population size, crossover rate, mutation rate, how you decided to terminate. Specify the α and β used in your SA, etc.

5. Describe your datasets
   Describe your toy (contrived) dataset and what the optimal solution is, and if your algorithms were able to obtain the optimal or not. This is very important. You need to convince me that your GA and SA actually work. To do this you need to run a toy problem where the optimal is known and your algorithms (GA and SA) should obtain the optimal. Describe your other datasets (small, medium and Large), where you got them and if the optimal is known.

6. Computational results.

Describe your results Use Tables, charts and/or graphs or whatever you like. Your report must include a Table depicting your results in a clear an easy to read form. For example, model your table(s) of results after the example table for the TSP on the next page. At the very least you should include at least one table per dataset (toy, medium size and large size). Clearly indicate what combinations of reproduction-crossover-mutation operators seems to work best for your problem. Highlight of otherwise indicate the best solutions on each of the tables.

7. Conclusions

What are your conclusions? Which technique seems to work best and what parameters. Did you find any surprises or special difficulties while developing the software. Any special conclusions or observations? What if anything did you personally learn from this project. Any other comments? Perhaps future ideas or techniques to try. In your report be sure to include what conclusions you came to comparing your results from the SGA, SA, and "Foolish" Hill Climbing with respect to crossover/mutation/perturbation functions.

8. References (if any)

Due date for all three parts of the written project is Monday April 24, 2023.

Late penalty is 2% per day

Oral PowerPoint presentations will begin Wednesday April 19, 2023.

Example Table of Results for the TSP
(Each program was run 5 times)
Adapt this table for your Particular Problem

Dataset Name/Size:____ Optimal value (if known) _________

		Best Tour Length	Average Number of Generations/Perturbations	Average Best Tour Length
SGA	Selection I Crossover I Mutation I			
	Selection I Crossover I Mutation II			
	Selection I Crossover II Mutation I			
	Selection I Crossover II Mutation II			
	Selection II Crossover I Mutation I			
	Selection II Crossover I Mutation II			
	Selection II Crossover II Mutation I			
	Selection II Crossover II Mutation II			
SA	Perturbation I α = β = T_0 = I_0 =			
	Perturbation II α = β = T_0 = I_0 =			
Foolish	"Foolish" Hill Climbing I			
	"Foolish" Hill Climbing II			

Rubric used by the Instructor to evaluate your Software Project and Report
CS 4623/6623 Evolutionary Computation
Wainwright

Software Project
Evaluation Criteria

NAME ____

Suggested paper outline:
1. Introduction--Clearly describe the problem you are solving.
2. Chromosome and fitness (clearly describe both)
3. Operators---Clearly describe the reproduction (selection) operators
4. Parameters used---Clearly specify your GA parameters
5. Describe your datasets
6. Computational results.
7. Conclusions
8. References (if any)

DETAILS
Description and identification of the problem to be solved________________________

Part I SGA

Description of the Simple Genetic Algorithm________________________________

At least two different selection techniques____________________________________

At least two different crossover operators____________________________________

At least two different mutation ____

The crossover rate____

The mutation rate____

Population size and other GA parameters____________________________________

The representation for the chromosome (bit string, permutation of integers, etc.)_______

The fitness function____

The population size____

Elitism must be implemented____

Describe how the GA terminates____

Generational GA must be used____

Specify how the initial population is determined____________________________________

Describe how you obtained your datasets____

Part II
Simulated Annealing Algorithm ____

Describe the two perturbation functions____

Part III
a "Foolish" Hill Climbing Algorithm____

Conclusions

Description of datasets____

Was one dataset contrived____

Evidence that the software works:
Description of the TOY problem ____

Was optimal solution found on the TOY or contrived dataset?__________________________

Describing how your GA, SA, Foolish performed ____________________________________

Table depicting all of the results.____

Conclusions and anything specific that was learned or was a surprise._____________________

Any other comments____

Instructor Comments