ISE 5113 Advanced Analytics and Metaheuristics
Homework #7

Instructor: Charles Nicholson

Due: See course website for due date

Requirement details

1. Homeworks are to be completed in teams (see Canvas for details). If team members disagree on an answer, you can record solutions corresponding to each member (please clearly mark which solution belongs to which team member).
2. Your primary submission will be a single Word or PDF document and must be of professional quality: clean, clear, concise, yet thoroughly responsive.
3. Any code (e.g., Python) must also be submitted separately. Your code MUST be well-documented. Failure to submit the files will result in a penalty.
4. You cannot use preexisting Python packages for heuristics or metaheuristics. In fact, other than `numpy, copy, random`, and other basic utilities, you should seek specific permission from the instructor if you have any doubt. That is, *you* are responsible for creating the logic yourself.

In this assignment you will modify some provided Python code to implement a Simulated Annealing algorithm and also a Genetic Algorithm to solve the same instance of the knapsack problem. After implementing the code and solving the problem, please summarize your results in a table similar to Table 1.

Table 1: Example of results summary (numbers are not realistic)

Method								
SA	t_0	Cooling, t_k	M_k	# of temps	Iterations	Items selected	Weight	Value
	1000	$\frac{t_0}{1+0.9k}$	10	900	2102	87	320	3180
	800	$0.99t_{k-1}$	50	40	5333	27	230	1284
	1200	$0.99t_{k-1}$	50	40	5333	13	1250	2002
GA	Generations	Pop size	Crossover	Mutation	Elitism	Items selected	Weight	Value
	1000	250	0.8	0.1	top 3	23	2650	1921
	500	150	0.9	0.2	top 10%	39	1650	3914
	etc.							

Knapsack Problem Definition Given n different items, where each item i has an assigned value (v_i) and weight (w_i), select a combination of the items to maximize the total value without exceeding the weight limitations, W, of the knapsack.

IMPORTANT!: When generating random problem instance set $n = 150$ and use the provided seed value for the random number generator. The max weight is 2500.

Question 1: Simulated Annealing (40 points)

Using the provided Python code from the previous homework as a base, implement Simulated Annealing in code. Please address the following by including the associated *Python code excerpts* (as appropriate) and explanation of the code in the PDF file:

- Logic to determine the initial temperature
- At least two different temperature cooling schedules (the temperature update procedure) and explore some options for the number of iterations performed at a given temperature (M_k)
- Python logic excerpt for computing the probabilities of accepting a non-improving move
- Stopping criterion

Apply the code to the random problem instance and determine the best solution and objective value using the multiple variations of your algorithm (e.g., different cooling schedules, different starting temperatures, different values for M_k).

Question 2: Genetic Algorithm (60 points)

Using the genetic algorithm Python code base provided as a starting point, implement a genetic algorithm to solve the knapsack problem. Please address the following by including the associated *Python code excerpts* (as appropriate) and explanation of the code in the PDF file:

- createChromosome() logic
- crossover() logic
- Logic to compute chromosome fitness, e.g., any modifications to the evaluate() function
- rouletteWheel() logic
- mutate() logic
- insert() logic

Apply the technique to the random problem instance and determine the best solution and objective value using your revised algorithm. Test multiple versions of the GA (e.g., different number of generations, population sizes, crossover rates, etc.)