UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Engineering, Built Environment and IT
Department of Computer Science

COS 314

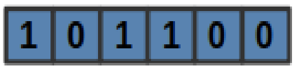
Tutorial/Practical 4 Solutions

24 March 2022

Questions

1. The one dimensional bin-packing problem involves assigning items to bins so as to minimize the number of bins used. Each of the bins has the same capacity and the sizes of the items are known. Suppose that you are required to implement a genetic algorithm to solve this problem.
   (a) Describe what each chromosome will represent.
   Sol: a number of solutions are possible here.
   The chromosome can represent the bins where each gene represents a bin. Then the sequence of the objects represents the bin it is placed in. E.g |3|2|5|6|1|4| (chromosome of length 6) means item/object 1 is placed in bin 3, item 2 in bin 2, item 3 in bin 5 and so on.
   (b) Define the fitness function that will be used.
   Sol: This fitness function must be an evaluation of using as few bins as possible and minimizing the empty(unused) space within each bin. F = sum of U^2/B for all the bins, where U is utilization and B is the number of bins [1]
   (c) Describe the selection method that will be used.
   Sol:Tournament selection
   (d) Describe at least two genetic operators that will be used.
   Sol: single point crossover and bit(gene) mutation (care must be taken as to avoid duplication this may involve swapping positions additionally it must result in feasible solutions.

2. Using the diagram given below as the starting individual develop implementations that apply the following local single point searches.
   (a) Simulated Annealing.
   (b) Tabu Search.



1	0	1	1	0	0

Figure 1:

parameter	value
population size	15
selection method	Tournament selection(size 4)
single-point crossover rate	85%
Bit mutation rate	15%
fitness function	number of 0 bits
maximum generations	50

Table 1: GA parameter settings

3. Implement a GA that searches for an 8-bit binary string of 1's. The GA should be configured with the parameters in Table 1. What effect will be the following have:

   - adjusting (high and low) the population size.
     Sol: a high population size may lead to quick convergence A low population size leads to the algorithm failing to converge or slow convergence
   - adjusting (high and low) the crossover rate.
     Sol: high crossover may take longer to converge or converge to a local optima as the algorithm will perform more exploitation than exploration. Low crossover slow convergence.
   - adjusting (high and low) the mutation rate.
     Sol: High mutation the algorithm becomes a random search while a very low mutation rate reduces diversity and the search is localised.
   - adjusting (high and low) the number of generations.
     Sol: high generations may lead to overfitting depending on the task. Low generations the algorithm may not converge

References

1. Bennell, J.A., Lee, L.S. and Potts, C.N., 2013. A genetic algorithm for two-dimensional bin packing with due dates. International Journal of Production Economics, 145(2), pp.547-560.