

# Advanced Nature Inspired Search and Optimisation

CA 2: Airline crew scheduling using GA

1429527

December 8, 2016

## 1 The Algorithm

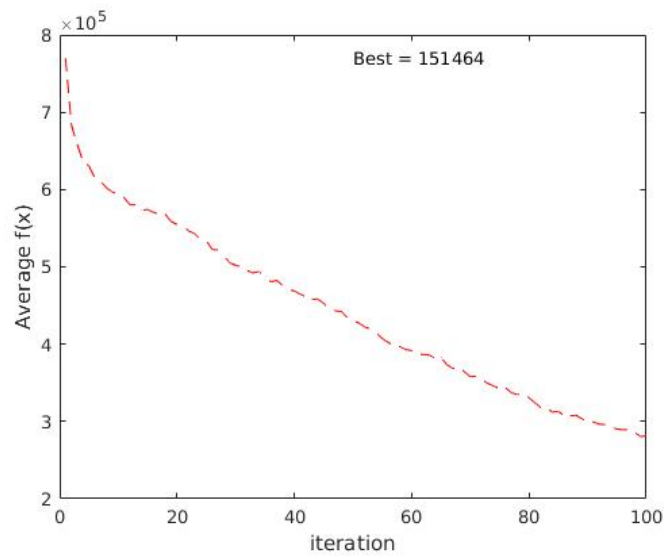
This genetic algorithm will be used to solve the 'Airline Crew Scheduling' problem where the cost needs to be minimised while traveling each leg of the journey.

To implement the algorithm, I used the code given [3] as the basis for my algorithm and added the required functionality on top.

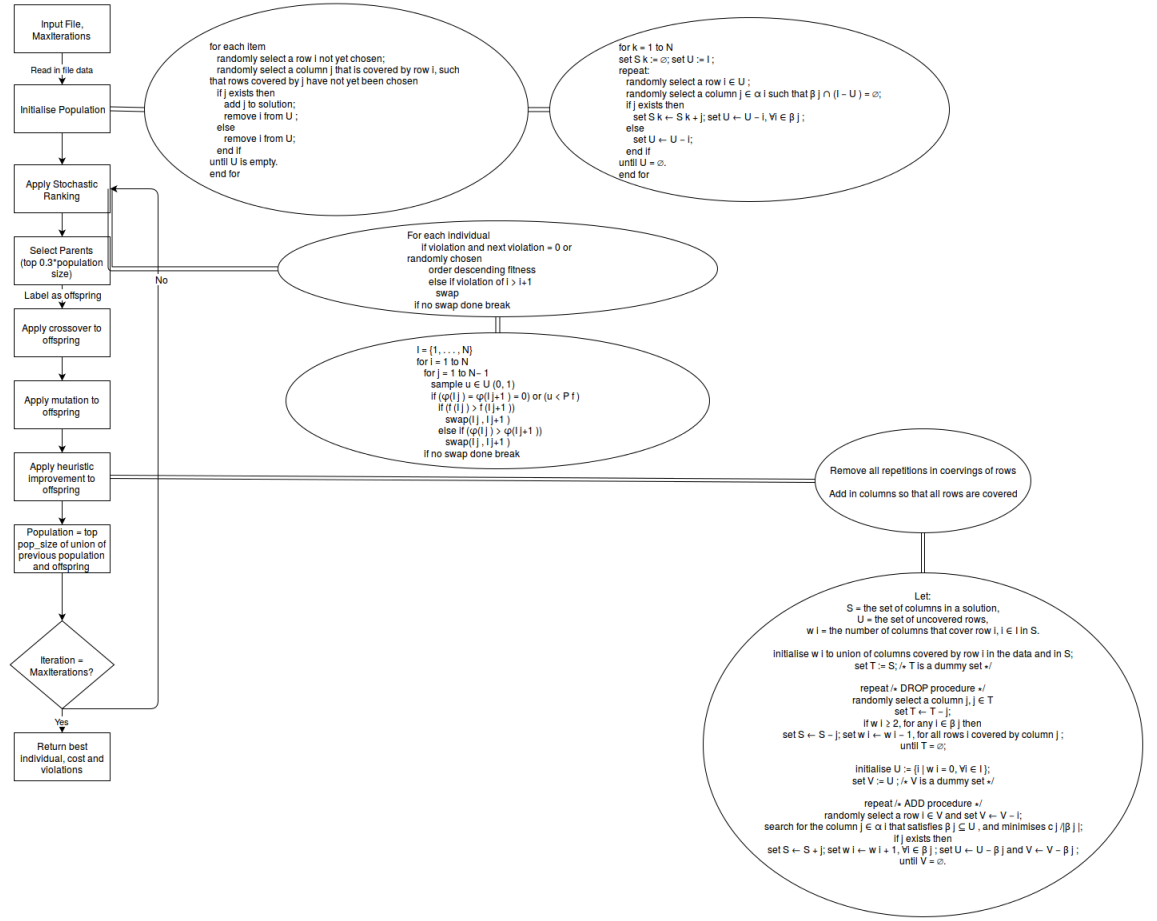
The basis of the algorithm is to randomly initialise the population of a specified size, then use the top parents to produce offspring. Stochastic Ranking was used to find the fittest parents, then crossover, mutation and heuristic improvement was applied to produce the offspring. Heuristic improvement changed any columns which can be changed - i.e. more than one row is covered by the currently selected row, then filling in any gaps in the solution with minimal cost. Finally the top population size number of individuals from the original population and the offspring are selected to be the new population.

The Stochastic Ranking not only ranks by fitness, but penalises any violation by a set amount. It also introduces randomness in that swaps can occur if a value is higher than a threshold, without the penalty being smaller.

Below is a graph plotting the current, best solution against the iteration number for benchmark sppnw41.



Below is the flow chart for the algorithm, containing sudo code given in [1] and [2].



## 2 Benchmark Problem

Number of runs: 10

Population size: 200

Max Iteration: 1000

Violation penalty = 0.45

## 2.1 sppnw41.txt

Run	Min Cost	Constraint Violations
1	13218	0
2	14253	0
3	12663	0
4	13209	0
5	13572	0
6	12663	0
7	11679	0
8	12897	0
9	13663	0
10	14253	0

Average Result: 13204

Average Violation: 0

Standard Deviation: 785.4421

## 2.2 sppnw42.txt

Run	Min Cost	Constraint Violations
1	11598	1
2	12210	0
3	16656	1
4	11544	0
5	12114	2
6	11260	1
7	11828	1
8	11430	0
9	6940	2
10	11272	0

Average Result: 11685

Average Violation: 0.8

Standard Deviation: 2313

### 2.3 sppnw43.txt

Run	Min Cost	Constraint Violations
1	8890	1
2	12018	0
3	11330	0
4	14076	0
5	10650	0
6	11036	0
7	10912	0
8	12090	0
9	14576	0
10	14666	0

Average Result: 12024

Average Violation: 0.1

Standard Deviation: 1889

## 3 Ranking Comparisons

[1] uses a Ranking Replacement strategy to order and ultimately discard unfit individuals whereas the above, along with [2] uses Stochastic Ranking.

Ranking replacement considers both the fitness (quality) and unfitness (feasibility) of the solutions. It divides the current population into four groups compared to the offspring: G1 less fit and more unfit, G2 more fit and less unfit, G3 less fit and more unfit and G4 more fit and less unfit. The offspring then replaces the member of a group with the worst unfitness, in that order - if G1 is empty, it will replace an item in G2 and so on.

Stochastic Ranking only compares the fitness of individuals. Probability is introduced to allow infeasible options to be compared with those feasible; If both are feasible then the probability of comparing their fitness is 1 else it is  $P_f$ . This method comes from the need to balance the objective function and penalty function as so will most likely be more useful than Replacement Ranking if the penalty function is high. All items in the population and the offspring are compared in this way and then the bottom ranking (least fit) individuals are discarded so that the population remains at a constant size.

With both methods, care needs to be taken so that duplicate solutions are not entered as this limits the algorithm's ability to generate new solutions. However with Replacement Ranking, it appears that the offspring must always be entered into the new population even if it is worse than all items in the current population whereas Stochastic will reject all worse solutions regardless of when they were created.

Stochastic ranking will most likely be more computationally efficient as it will have the same complexity as bubble-sort on which it is based. Replacement

Ranking requires the entire population to be split into four sections with respect to the offspring trying to be replaced, meaning popSize computations for every offspring; This may be efficient if the number of offspring is small.

Replacement ranking dictates the item to be replaced by an offspring, however Stochastic ranking, as it has the variable  $P_t$ , can change the weight of infeasible solutions over time. This means that, as time goes on, the chance of an infeasible solution being carried forward can be reduced so as to focus on the current local optima. Ranking is designed to keep the scope of the solutions broad and does not have such a variable to create an adaptive algorithm.

## References

- [1] P.C. Chu, J.E. Beasley, Constraint Handling in Genetic Algorithms: The Set Partitioning Problem, *Journal of Heuristics*, 11: 323-357 (1998).
- [2] T.P. Runarsson, X. Yao Stochastic ranking for constrained evolutionary optimization, *IEEE Trans. on Evolutionary Computation*, 4(3): 284-294 (2000).
- [3] S. He. Week 6 Lab Code Examples. (2016)