Simulation - Assignment 3

Anton Roth

May 31, 2017

Implementation

All tasks have been solely implemented in *Matlab*. Files *main_ga.m* and *tsp_ga.m* contain the implementations.

The implemented parts of the *Genetic Algorithm* (GA) were cross-checked individually for satisfactory behaviour. The GA was validated through the following two tests:

1. Fitness value decreasing with increasing number of generations.

2. Fitness value decreasing with increasing population size.

The test were performed on the TSP with 48 cities with the simulation parameters given in the assignment instructions. The results are presented in Table 1 and 2.

Table 1: Fitness value vs. increasing number of generations. The population size was set to 100.

Number of generations	Fitness value
100	100000
1000	60000
2000	50000

Table 2: Fitness value vs. increasing population size. The number of generations was set to 1000.

Population size	Fitness value
10	140000
50	65000
200	50000

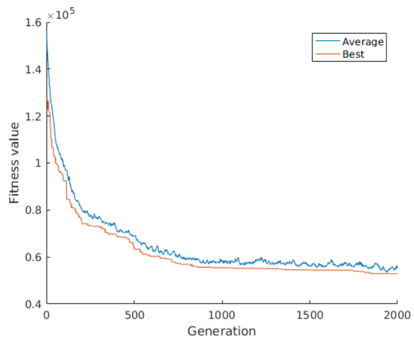
It was noted that the implemented GA did not perform as well as the algorithm used in lab 3. For a population size of 100 and 1000 generations it reached a fitness value of 34000 (cf. 60000 for this GA). This is troubling. However, it can be due to many reasons. For instance:

- The simulation parameters in this GA might not be as optimised as in the other GA.

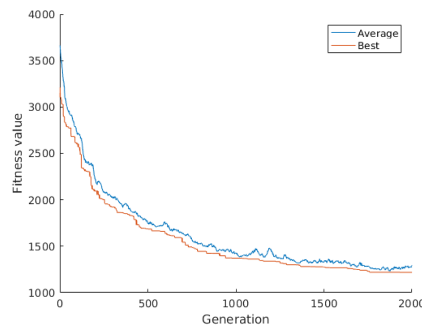
- The GA implementation itself might not be as optimal as in the other GA.

Task 1

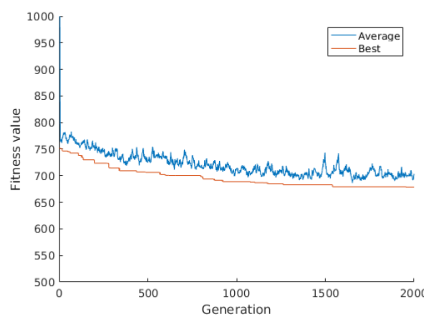
The results of the simulations are presented in Figure 1.



(a) 48 cities.



(b) 70 cities.



(c) 96 cities.

Figure 1: The implemented GA was run for three different TSPs. The population average and the best fitness value has been plotted vs. the population generation.

The fitness, of the population average such as the best in the generation, decreases as the generations pass. This indicates an improvement of the population in general and the best individual in the population. The slope of the curves decreases the further along the generations one goes. Thus, it can be interpreted that the population converges to a certain type of solutions (near optimal).

The average fitness follows the trend of the best fitness strictly, but with an added positive shift. Of course the average should be larger than the minimum of the population. Interestingly, the curve patterns are, if fluctuations are neglected, almost identical. This could be due to the *roulette wheel* selection in combination with the elitism which always works to favour the best fitted individuals.

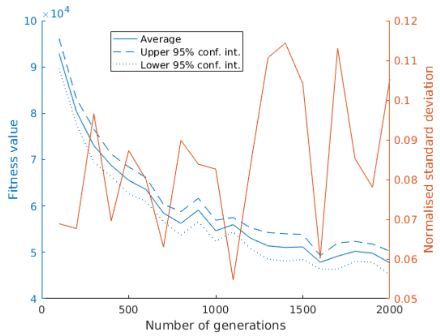
If one compares the three simulations it can be observed that the convergence is the fastest for the 96 city TSP (“converged” $\sim$ generation 500) . The convergence of the 48 city TSP (“converged” $\sim$ generation 1000) is faster than the 70 (“converged” $\sim$ generation 1600) city TSP. It seems logical that the more cities in the TSP the “farther” you can be from the optimal solution. Hence, a more rapid convergence should be pronounced. However, this is not observed comparing the 48 and 70 city TSPs. It could be that the structure of the nodes in the TSP plays a significant role. What effects the TSP structure have on the convergence is unclear for me though.

Task 2

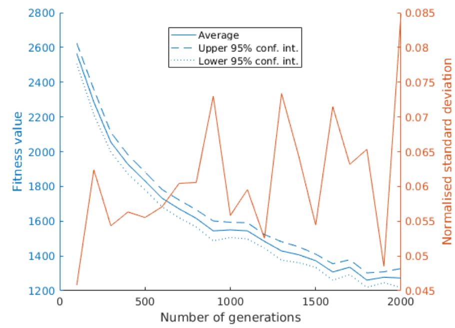
The implemented GA was run 15 times for a total number of generations ranging from 100 to 2000 with a step size of 100. For each total number of generation the average of the best fitness value and its upper and lower 95% confidence limits were calculated. The 95% confidence interval was calculated using the normal distribution approximation as:

$$[\bar{x} - 1.96 \cdot \frac{\sigma}{\sqrt{N}}, \bar{x} + 1.96 \cdot \frac{\sigma}{\sqrt{N}}]$$

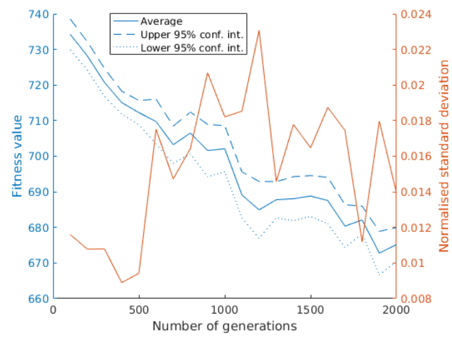
Here x is the samples, $\bar{x}$ the estimated mean, σ the sample standard deviation and N the number of samples. The results of the simulation of task 2 is presented in Figure 2.



(a) 48 cities.



(b) 70 cities.



(c) 96 cities.

Figure 2: The average of the best fitness value (left y-axis) is presented with its upper and lower 95% confidence limits for the 20 different total number of generations. On the right y-axis the normalised standard deviation (w.r.t. average best fitness value) per total number of generations is shown. The three different TSPs in (a), (b) and (c).

What is clearly observed in each of the plots in Figure 2 is the improvement of the final value of the GA algorithm with increasing total number of generations simulated. This is expected since the roulette wheel selection and the elitism works to always keep the best solutions in the population. The crossover and the mutation enables the improvement. The improvement can be observed to be the largest for a smaller total number of generations simulated. It can be explained by that there is always the largest potential improvement of the first generation while it decreases as generations pass since the optimal solution is approaching.

The convergence speed, i.e. after how many generation does not the fitness value improve, is somewhat difficult to compare between the three TSPs. However, it seems that the 48 and 70 city TSPs flattens rather nicely at around 2000 generations. The 96 city TSP seems to need more than 2000 generations to converge.

Interesting to note is that the convergence of the 96 city TSP assumes a more linear behaviour while the others assume more an exponential behaviour.

Another interesting feature is that the curves are not strictly decreasing. At some points one can see “bumps”. But this is natural, since the GA is a probabilistic process sometimes it might happen that the population does not get the improving mutation or crossover as it did in other runs.

From the red curves, i.e. the normalised standard deviation as a function of total number of generations, in Figure 2 one can conclude that the confidence interval is the tightest for the 96 city TSP. This means that the best value after each simulation does not deviate here as much as for the other TSPs. If one studies the pure standard deviation (can also be seen on the confidence intervals) it is difficult to obtain any pattern. It seems to fluctuate around some mean value. This is interesting in itself and might cause you to draw the conclusion that how certain you can be of the obtained “optimal” solution does not depend on the number of iterations computed. But, I would say that several more simulations per number of iterations are required to draw any good conclusions.