

IT3708 Project 5

Solving Multi-Objective Traveling Salesman Problem (MTSP) using Multi-Objective Evolutionary Algorithm

Iver Jordal and Olav Markussen

Description of the complete MOEA design

Our genotype representation is a list of cities in the visited order. This made crossover and mutation easy to implement. The crossover function takes two genotypes, directly copying an interval from one of them into the resulting genotype, while looping through the second genome and inserting genes where it is possible. The mutation function simply swaps two random cities with each other. Both crossover and mutation satisfies the TSP requirements.

Each genome, individual and population are classes, where a population contains several individuals, and one individual contains one genome. The genome class is responsible for initializing the genotype, as well as crossover and mutation functions. The individual handles everything concerning calculation of costs. This includes tour cost, tour distance and crowding distance. The population takes care of finding fronts (fast dominated sort) and selecting parents (tournament selection). In accordance with NSGA-II, we use tournament selection, as our selection strategy. It defines two random contestants from the whole population, selecting the best one (first sort on rank, then on crowding distance) with a high probability, and selecting the other with a low probability. The controlling part is in our main function, that sets all parameters, and loops through the generations, updating the generation in accordance with NSGA-II, until the generation limit is reached.

Discuss mutation and crossover functions

As our mutation function is pretty straight forward, only changing the permutation of the genome, we don't see it producing an infeasible solution. It will never produce a list with duplicate values for instance. We neither see the crossover producing an infeasible solution. It simply directly copies a route from one parent into the resulting genome. Afterwards it adds the missing route locations in the order of the second parent. The resulting crossover method didn't add nor subtract any cities to the result, it only changed the permutation of the existing two genotypes. The method is easily seen in the figure below. The first two lists are the parents. The last two lists show the genome of the child in each stage of the crossover function.

1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---

9	8	7	6	5	4	3	2	1
---	---	---	---	---	---	---	---	---

					6	7	8	
--	--	--	--	--	---	---	---	--

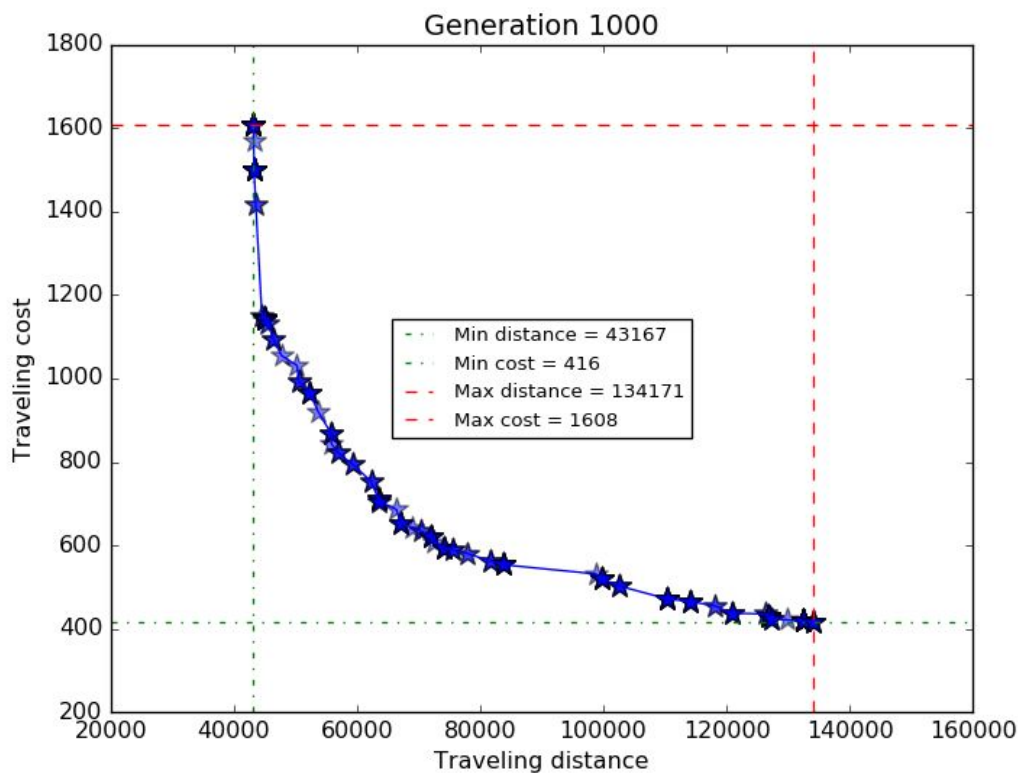
9	5	4	3	2	6	7	8	1
---	---	---	---	---	---	---	---	---

Document your EA choices

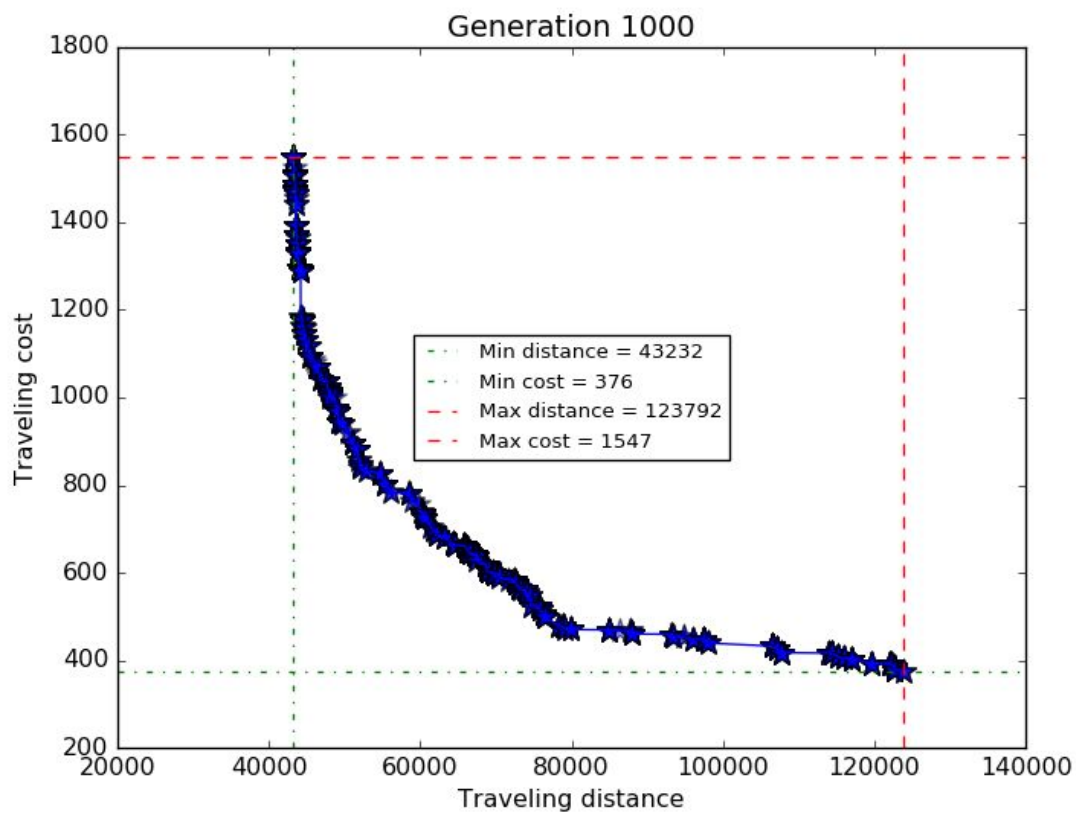
Population size	# generations	crossover rate	mutation rate	Tour Distance (best, worst)	Tour Cost (best, worst)	Non-dominated solutions
200	1000	0.8	0.4	43167, 134171	416,1608	200
1000	1000	0.5	0.5	43232, 123792	376, 1547	1000
400	2000	0.5	0.5	39568, 126217	359, 1732	400

Plotting of final solutions

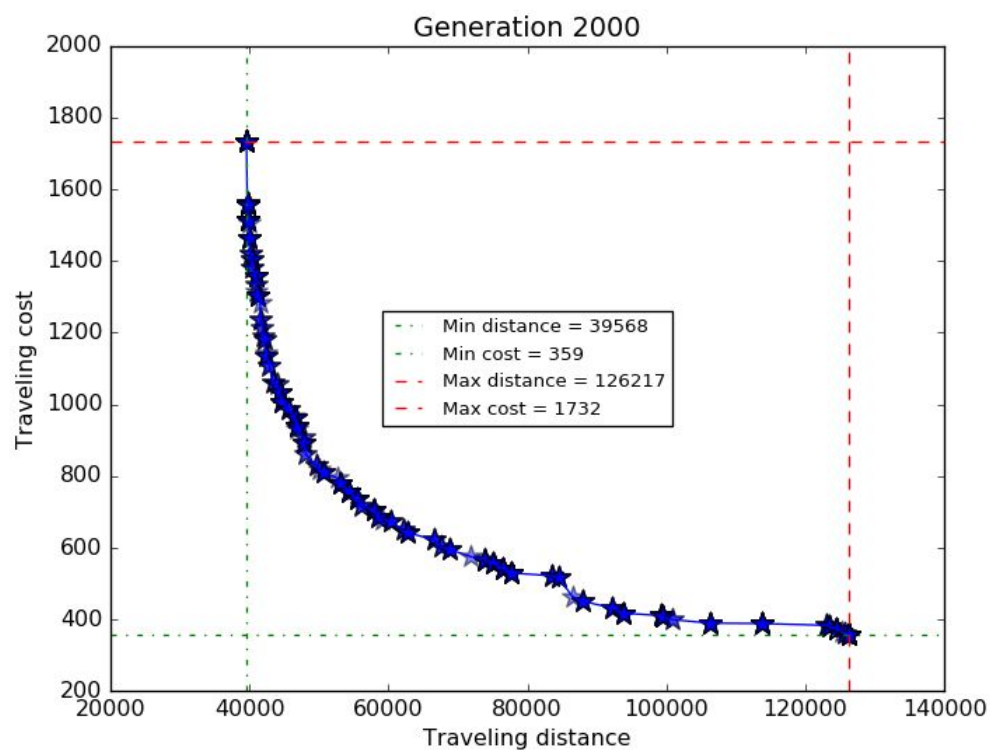
The following plots are the final plots for each set of parameters in the table above. We believe that the following four plots satisfy the conditions given in the assignment. The reason being that we are to “to plot every members of the final generation” in task 5.3.1 and in task 5.3.2 we are plotting in accordance to “the members will be only the non-dominated solutions”. In our solutions, every member of the final non-dominated solutions is the final generation. This means that the pareto-front consists of all members of the generation. Because of this, we have only included four plots, since it is redundant to show the same plot twice, with different meaning.



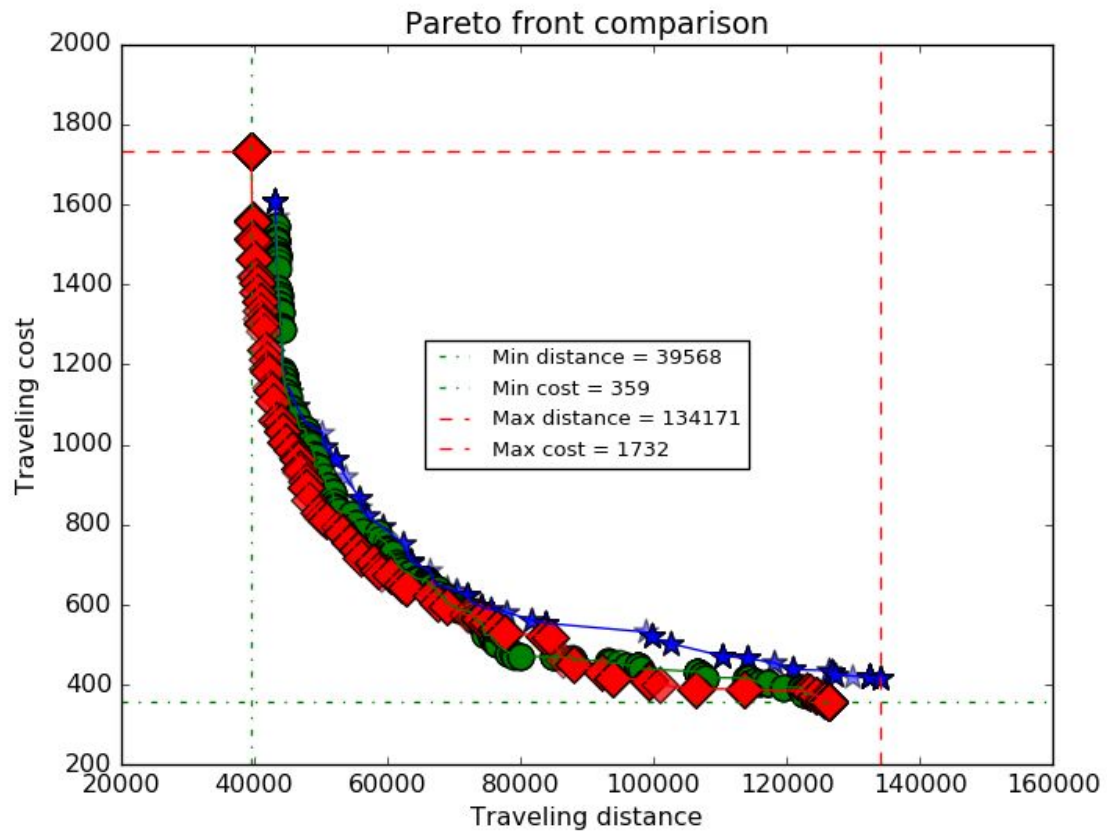
Set 1



Set 2



Set 3



Comparison between all three fronts. Set 1 is blue (stars), set 2 is green (circles) and set 3 is red (diamonds).