Travelling Salesman Problem simulated with Simple Genetic Algorithm

Kevin Tram
Computer Science
Brock University
St Catharines, Ontario Canada
Student Number: 5459854
Email: kt13nh@brocku.ca

***Abstract*—Travelling salesman problem, a very well known problem among the computer science community, tries the solve this problem with a variety of different strategies and algorithms. The problem is this: if a salesman must visit all the cities in a given place, what is a good route that the salesman can follow to efficiently travel to visit all of the cities? At first glance, one can imagine with a small data set, it can be relatively easy to determine the shortest path by simple trial and error. However, as we're going to see with "berlin52" and "djibouti38" data set, this is not the case. In this report, we will examine these two data sets and observe the significant impact genetic algorithms can have on the solutions of each of these data sets. Each solution is a representation of a route, where experiments will be conducted in an attempt to solve the TSP problem with a reasonably good solutions with two different crossovers, and one type of mutation. It is observed that Order Crossover generally performed better than Uniform Order Crossover.**

I. Introduction

The Travelling salesman problem was first introduced in 1930 and since, one of the most studied optimization problems by scholars. This is an NP-Hard problem used as a benchmark for many different optimization methods, genetic algorithms included. The experiment will look at how different genetic algorithms including a variety of crossover methods, mutations, and parameters will affect the outcome best route distances of every generation. These different methods of crossover and parameters will be compared against each other in terms of performance, examined to determine which will yield the best possible solutions for each of the datasets provided.

A. *Objective and problem definition*

With the provided data sets "Berlin52", and "Djibouti38" a possible good solution is found by implementing a simple genetic algorithm incorporating a variety of different user parameters for crossover rates, mutation rates, and types of crossovers. This main problem is defined as: a salesman must travel to all of the cities in a specific area at least once. This problem represented based on genetic algorithms using chromosome solutions as cities, as well as always ensuring the chromosomes are valid tsp solutions. Each chromosome is represented as a set of genes, and each gene is represented as a city, where each collection of genes is representation for a valid solution. Every city contains relevant attributes, such as its X-coordinate, and Y-coordinate. Distance among city to city is calculated using the well known Euclidean distance illustrated below:

1) Euclidean Distance Equation:

$$d = \sqrt{((x_1 - x_2)^2 + (y_1 - y_2)^2)} \tag{1}$$

After new generation productions, the hope is to develop parent solutions into better solutions using an elite strategy where the best chromosome from the previous generation is always going to be included in the next generation. We will analyze the solutions being produced by the simple genetic algorithm with Uniform Order Crossover and Order Crossover. Inversion mutation will also have a chance to be applied to each of the genes in the next generation as entered by the parameters. Tournament selection will be used for all of the new generations where k-value 3 is used to determine which chromosome is to make it through the tournament. Once selection is complete, crossover and mutations are applied to the new offspring. After producing many generations, and inheriting the best chromosome after every generation, the difference in fitness of the chromosomes produced in the genetic algorithm will be observed, and used determine the best user parameter settings, with crossover and mutation rates. This will be achieved with the following simple GA algorithm:

Algorithm 1 Simple GA:

```
procedure SIMPLE GA
    Begin
    Read problem instance data;
    Generate random initial POP of size PopSize;
    for gen = 1 to MAXGEN do
        Evaluate fitness of the individuals of POP;
        Select new POP using tournament selection;
        Apply genetic operators, crossover, mutation;
    end for
    end;
end procedure
```

B. *Summary Parameters Used*

1) Berlin52 Uniform Order Crossover: The following parameters are used in the experiment of Berlin52 using Uniform order Crossover and Inversion Mutation:

Berlin52 Uniform Order Crossover	
Crossover Rate used	1
Mutation Rate used	0
Population Size	52
Seed used	55
Max generation	10000
Best Fitness	14167.06283

Berlin52 Uniform Order Crossover	
Crossover Rate used	1
Mutation Rate used	0.1
Population Size	52
Seed used	55
Max generation	10000
Best Fitness	7643.64272

Berlin52 Unifrom Order Crossover	
Crossover Rate used	0.9
Mutation Rate used	0
Population Size	52
Seed used	55
Max generation	10000
Best Fitness	15796.02808

Berlin52 Uniform Order Crossover	
Crossover Rate used	0.9
Mutation Rate used	0.1
Population Size	52
Seed used	55
Max generation	10000
Best Fitness	8060.575782

Berlin52 Uniform Order Crossover	
Crossover Rate used	1
Mutation Rate used	0.2
Population Size	52
Seed used	55
Max generation	10000
Best Fitness	7558.001888

As easily observed in the above parameter selections, the crossover rate of 1 and mutation rate of 0.2 yielded the most efficient solution for the Berlin52 dataset using Uniform Order Crossover and maximum generation 10,000.

2) Berlin52 Order Crossover: The following parameters are used in the experiment of Berlin52 using Order Crossover and Inversion Mutation:

Berlin52 Order Crossover	
Crossover Rate used	1
Mutation Rate used	0
Population Size	52
Seed used	55
Max generation	10000
Best Fitness	14561.96866

Berlin52 Order Crossover	
Crossover Rate used	1
Mutation Rate used	0.1
Population Size	52
Seed used	55
Max generation	10000
Best Fitness	7545.713105

Berlin52 Order Crossover	
Crossover Rate used	0.9
Mutation Rate used	0
Population Size	52
Seed used	55
Max generation	10000
Best Fitness	15079.99702

Berllin52 Order Crossover	
Crossover Rate used	0.9
Mutation Rate used	0.1
Population Size	52
Seed used	55
Max generation	10000
Best Fitness	7979.856824

Berlin52 Order Crossover	
Crossover Rate used	0.9
Mutation Rate used	0.2
Population Size	52
Seed used	55
Max generation	10000
Best Fitness	7350.876637

As observed in the above tables of Order Crossover experimented with the Berlin52 dataset, crossover rate of 0.9 and mutation rate of 0.2 yielded the most optimal fitness value with maximum generation 10,000.

3) Djibouti38 Uniform Order Crossover: The following parameters are used in the experiment of Djibouti38 using Uniform Order Crossover and Inversion Mutation:

Djibouti38 Uniform Order Crossover	
Crossover Rate used	1
Mutation Rate used	0
Population Size	38
Seed used	55
Max generation	10000
Best Fitness	12850.31611

Djibouti38 Uniform Order Crossover	
Crossover Rate used	1
Mutation Rate used	0.1
Population Size	38
Seed used	55
Max generation	10000
Best Fitness	6352.410452

Djibouti Uniform Order Crossover	
Crossover Rate used	0.9
Mutation Rate used	0
Population Size	38
Seed used	55
Max generation	10000
Best Fitness	16519.46246

Djibouti Uniform Order Crossover	
Crossover Rate used	0.9
Mutation Rate used	0.1
Population Size	38
Seed used	55
Max generation	10000
Best Fitness	6729.912659

Djibouti Uniform Order Crossover	
Crossover Rate used	1
Mutation Rate used	0.3
Population Size	38
Seed used	55
Max generation	10000
Best Fitness	6322.798899

As observed in the above tables of Uniform Order Crossover experimented with the dataset Djibouti38, a crossover rate of 1 and mutation rate of 0.3 yielded the most optimal fitness value with maximum generation 10,000.

4) Djibouti38 Order Crossover: The following parameters are used in the experiment of Djibouti38 using Uniform Order Crossover and Inversion Mutation:

Djibouti Order Crossover	
Crossover Rate used	1
Mutation Rate used	0
Population Size	38
Seed used	55
Max generation	10000
Best Fitness	12989.16686

Djibouti Order Crossover	
Crossover Rate used	1
Mutation Rate used	0.1
Population Size	38
Seed used	55
Max generation	10000
Best Fitness	6522.474191

Djibouti38 Order Crossover	
Crossover Rate used	0.9
Mutation Rate used	0
Population Size	38
Seed used	55
Max generation	10000
Best FItness	15123.09153

Djibouti38 Order Crossover	
Crossover Rate used	0.9
Mutation Rate used	0.1
Population Size	38
Seed used	55
Max generation	10000
Best FItness	6315.568273

Djibouti38 Order Crossover	
Crossover Rate used	0.9
Mutation Rate used	0.3
Population Size	38
Seed used	55
Max generation	10000
Best FItness	6300.17548

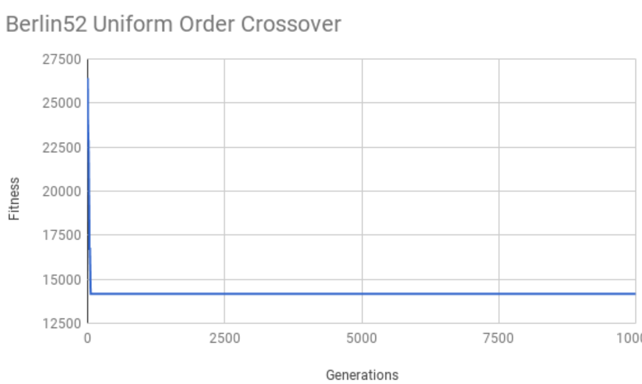
As observed in the above tables of Order Crossover experimented with the dataset Djibouti38 a crossover rate of 0.9 and mutation rate of 0.3 yield the most efficient chromosome fitness value with maximum generation of 10,000.

All of the user parameters used are consistent throughout all of the simulation data. Population size of the number of cities are used, as well as a seed of 55 and max generation are kept consistent among all of the simulations in an attempt to gather data to compare uniform order crossover with order crossover.

C. Results

Berlin52 Uniform Order Crossover a)

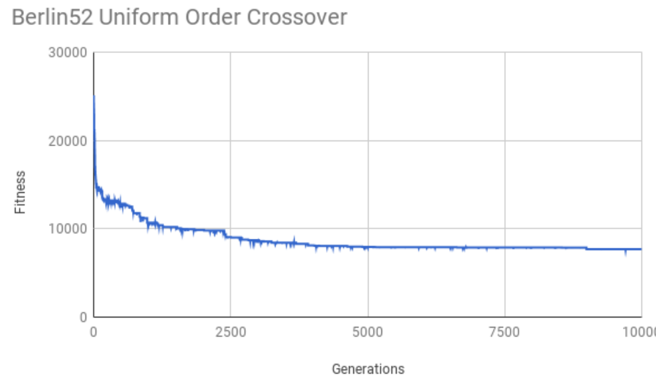
Berlin52 Uniform Order Crossover	
Crossover Rate used	1
Mutation Rate used	0
Population Size	52
Seed used	55
Max generation	10000
Best Fitness	14167.06283



From the above graph an the drastic slope in the beginning generations improves in fitness significantly. However, shortly after the earlier generations the fitness completely plateaus, indicating the limitations of a GA without mutation. It seems as though all of the chromosomes after the initial improvements in fitness suffer from farther improvements due to the lack of diversity in the population. Overall, these parameters have proven to not be useful in terms of evolving for better offspring.

Berlin52 Uniform Order Crossover b)

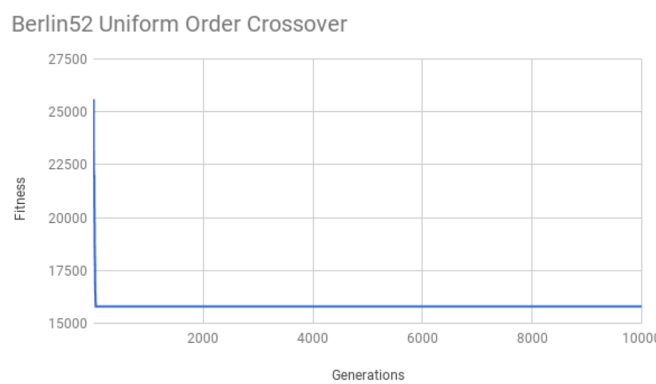
Berlin52 Uniform Order Crossover	
Crossover Rate used	1
Mutation Rate used	0.1
Population Size	52
Seed used	55
Max generation	10000
Best Fitness	7643.64272



The above graph with the specified parameters improved drastically in fitness from the early generations. They plateau slightly on the 5000 generations mark, however, fitness seems to improve as more generations are improved. A much improvement over graph a), where fitness improve greatly from increasing mutation rate to 0.1. This may indicate slight mutations in each generation will improve the chromosome offspring.

Berlin52 Uniform Order Crossover c)

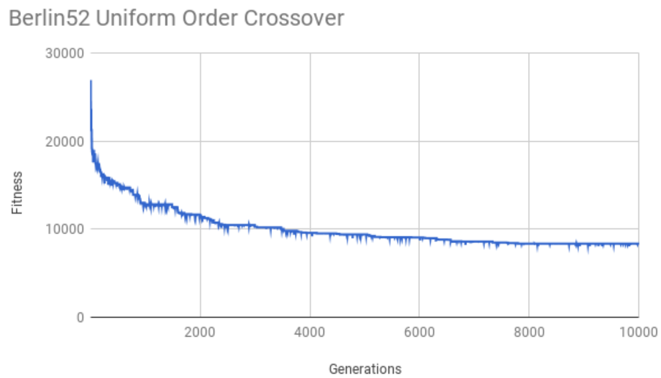
Berlin52 Unifrom Order Crossover	
Crossover Rate used	0.9
Mutation Rate used	0
Population Size	52
Seed used	55
Max generation	10000
Best Fitness	15796.02808



The above graph is much like graph a). There is a sheer improvement in the early generations, but plateaus quickly in a rather inefficient solution after 10,000 generations. With a best fitness value of less than graph a), this may also indicate chromosomes benefit from crossover where offspring will include the good parts of the parent chromosomes and carry forward to spawn even better offspring. These parameters have proven to be inefficient.

Berlin52 Uniform Order Crossover d)

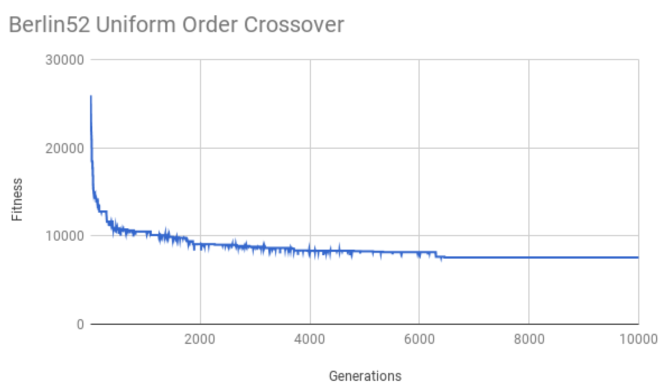
Berlin52 Uniform Order Crossover	
Crossover Rate used	0.9
Mutation Rate used	0.1
Population Size	52
Seed used	55
Max generation	10000
Best Fitness	8060.575782



The above graph shows constant signs of improvement, however, still not yielding fitness values as efficient as graph b). Since the crossover rate is less than b), there can be an assumption that higher crossover rates are beneficial to create more efficient children chromosomes than the parents, similar to the evidence presented from graph c).

Berlin52 Uniform Order Crossover e)

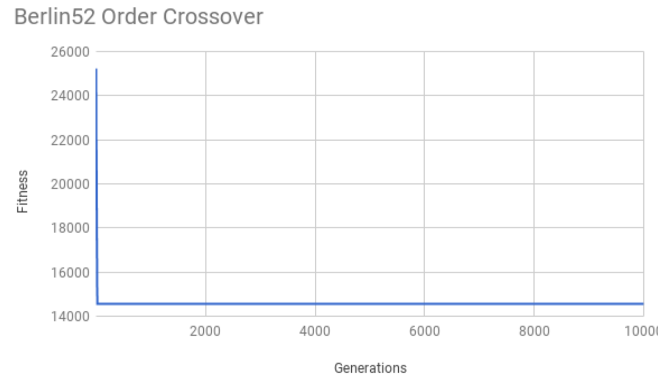
Berlin52 Uniform Order Crossover	
Crossover Rate used	1
Mutation Rate used	0.2
Population Size	52
Seed used	55
Max generation	10000
Best Fitness	7558.001888



From the evidence observed from graphs a) to d), a crossover rate of 1, and mutation rate of 0.2 is simulated based on the assumptions that high crossover rates are beneficial, as well as some mutations. The resulting fitness is an improvement over all of the Berlin52 Uniform Order Crossover graphs, and thus, the observations regarding chromosome fitness based on crossover rates and mutations are true.

Berlin52 Order Crossover a)

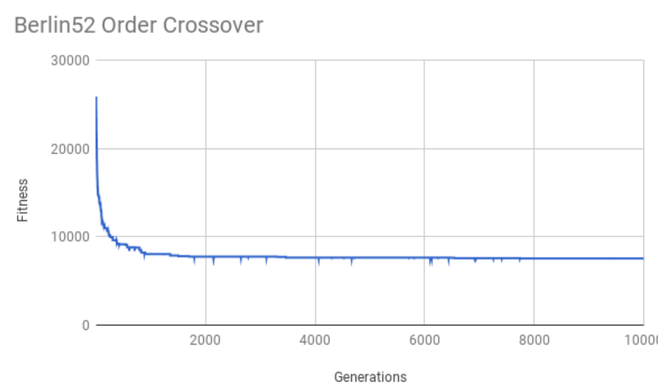
Berlin52 Order Crossover	
Crossover Rate used	1
Mutation Rate used	0
Population Size	52
Seed used	55
Max generation	10000
Best Fitness	14561.96866



There is a clear observation this set of parameters show very limited signs of improved chromosomes, and quick plateau that never improves. This is an obvious indicator this set will produce inefficient chromosomes as see by the plateau and the observed best fitness.

Berlin52 Order Crossover b)

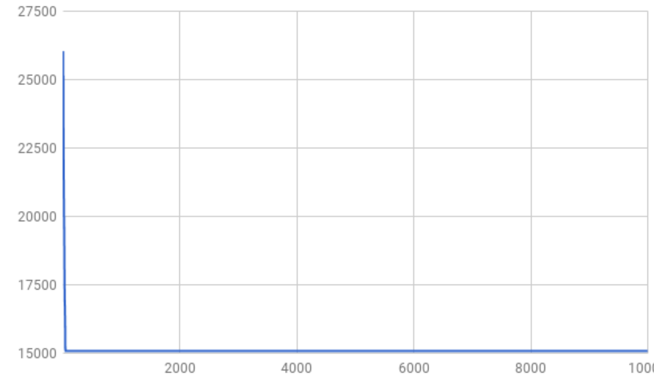
Berlin52 Order Crossover	
Crossover Rate used	1
Mutation Rate used	0.1
Population Size	52
Seed used	55
Max generation	10000
Best Fitness	7545.713105



As soon as mutation rate is increased from graph a), there is an immediate improve in performance. This is further indication that some mutations may yield increase of GA performance. Although plateau is evident past the 8000 generation mark, previous to that there are slight increases in performance throughout the generations.

Berlin52 Order Crossover c)

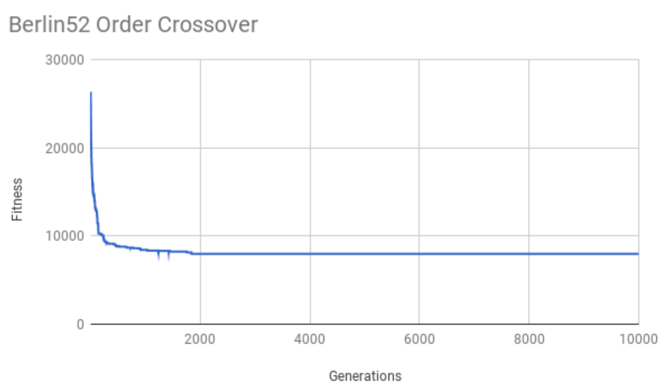
Berlin52 Order Crossover	
Crossover Rate used	0.9
Mutation Rate used	0
Population Size	52
Seed used	55
Max generation	10000
Best Fitness	15079.99702



The above graph is observed with less crossover rate than graph a), and with the same mutation rate, indicating that higher crossover rates will produce better offspring chromosomes. The sheer improvement in the early generations quickly plateau with no signs of improvement and a yield of a very low fitness value proves this set of parameters is inefficient.

Berlin52 Order Crossover d)

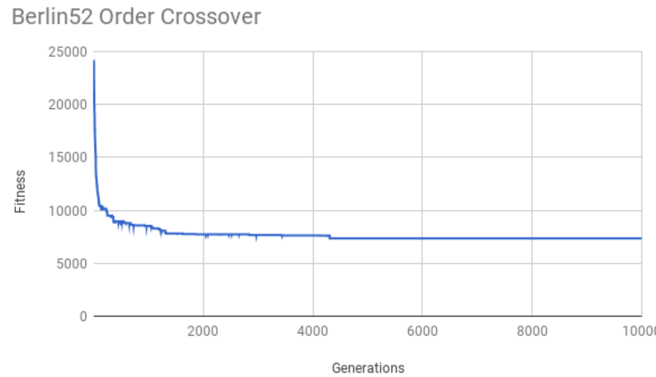
Berllin52 Order Crossover	
Crossover Rate used	0.9
Mutation Rate used	0.1
Population Size	52
Seed used	55
Max generation	10000
Best Fitness	7979.856824



The above graph with the same parameters as graph c), except for added mutation is a great increase in performance. Although plateaus, the offspring are close to 2 times better fitness than those of c).

Berlin52 Order Crossover e)

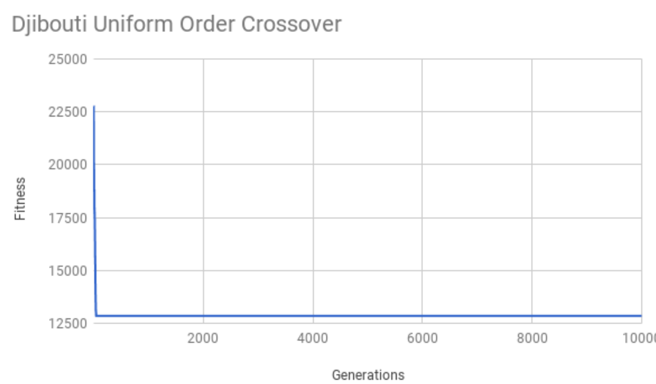
Berlin52 Order Crossover	
Crossover Rate used	0.9
Mutation Rate used	0.2
Population Size	52
Seed used	55
Max generation	10000
Best Fitness	7350.876637



With the observations from the Order Crossover, the parameters of crossover rate 0.9, and mutation rate of 0.2 is experimented. The Best fitness chromosome yielded better results than all of the observed Berlin52 graphs. Although tested with a 0.9 crossover, this may be indication Order Crossover is more efficient than Uniform Order Crossover on the dataset Berlin52.

Djibouti38 Uniform Order Crossover a)

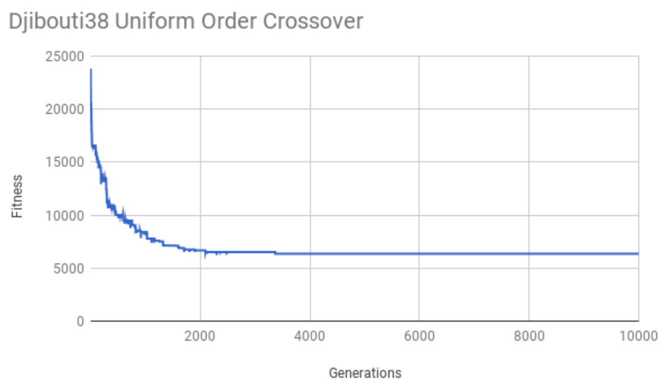
Djibouti38 Uniform Order Crossover	
Crossover Rate used	1
Mutation Rate used	0
Population Size	38
Seed used	55
Max generation	10000
Best Fitness	12850.31611



The above graph is consistent with the same parameters observed in the Berlin52 data set. There is a drastic improvement in fitness only to plateau quick and never improve again, yielding a very inefficient solution.

Djibouti38 Uniform Order Crossover b)

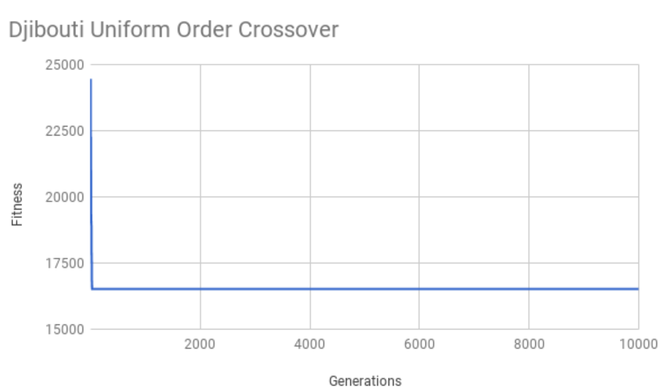
Djibouti38 Uniform Order Crossover	
Crossover Rate used	1
Mutation Rate used	0.1
Population Size	38
Seed used	55
Max generation	10000
Best Fitness	6352.410452



The mutation added, the graph is also consistent with the same parameters observed in Berlin52 where the graph will continually improve for several thousand generations until a plateau. This is yielding a good chromosome, though it seems to not improve after the plateau.

Djibouti38 Uniform Order Crossover c)

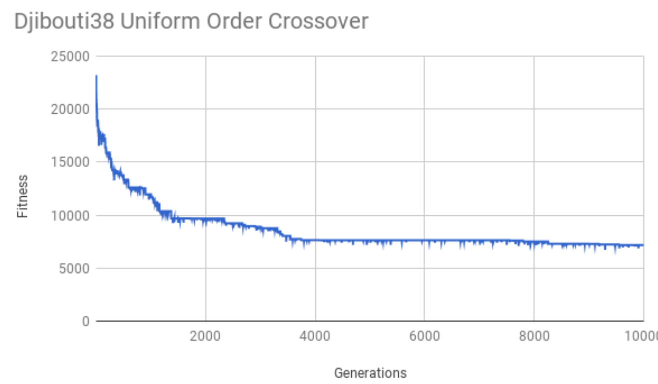
Djibouti Uniform Order Crossover	
Crossover Rate used	0.9
Mutation Rate used	0
Population Size	38
Seed used	55
Max generation	10000
Best Fitness	16519.46246



Much like graph a), this above graph yield an even worst solution with drastic beginning generation improvements, only to never improve again.

Djibouti38 Uniform Order Crossover d)

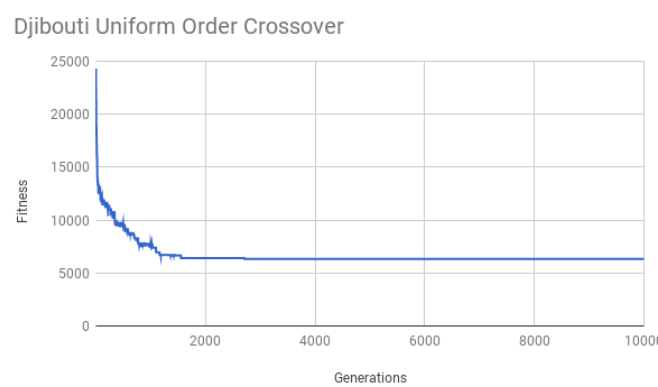
Djibouti Uniform Order Crossover	
Crossover Rate used	0.9
Mutation Rate used	0.1
Population Size	38
Seed used	55
Max generation	10000
Best Fitness	6729.912659



The graph above shows constant improvements in fitness, indicating the production of better offspring don't stop since it is observed the graph is jagged towards 10,000th generation, but showing no signs of complete plateau. Although this does not yield a better solution than graph b), graph b) is observed to plateau not not improve offspring again, with graph d) constantly improving chromosomes.

Djibouti38 Uniform Order Crossover e)

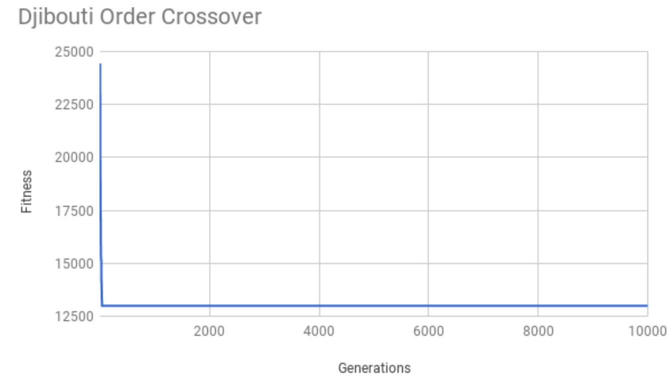
Djibouti Uniform Order Crossover	
Crossover Rate used	1
Mutation Rate used	0.3
Population Size	38
Seed used	55
Max generation	10000
Best Fitness	6322.798899



From all of the previous observations, an experiment was conducted on the basis that high crossover rates, and high mutation rate will yield good solutions. The graph seems to improve for over 1000 generations, and plateau after 2000. Although this yield better solutions than the other Djibouti Uniform crossover experiments, the plateau never seems to end and chromosomes never improve.

Djibouti38 Order Crossover a)

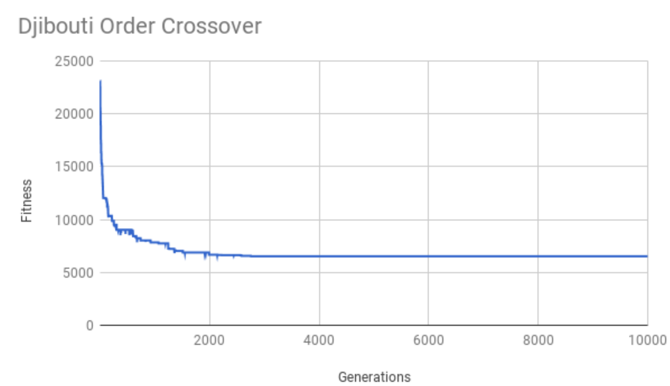
Djibouti Order Crossover	
Crossover Rate used	1
Mutation Rate used	0
Population Size	38
Seed used	55
Max generation	10000
Best Fitness	12989.16686



Drastic early generation improvement, also plateau early to never improve offspring again. This is inefficient as observed in all of the experiments conducted with this parameter setting.

Djibouti38 Order Crossover b)

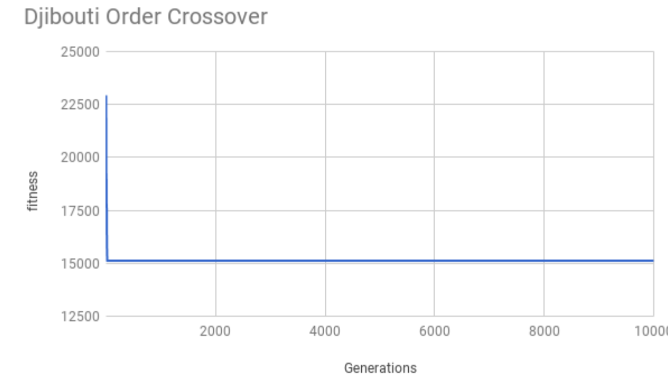
Djibouti Order Crossover	
Crossover Rate used	1
Mutation Rate used	0.1
Population Size	38
Seed used	55
Max generation	10000
Best Fitness	6522.474191



Very consistent with the other experiments conducted with the same parameter, it is observed that there is a continual improvement in offspring until slightly after the 2000th generation where it will plateau and offspring will not improve. However, still yielding a good solution.

Djibouti38 Order Crossover c)

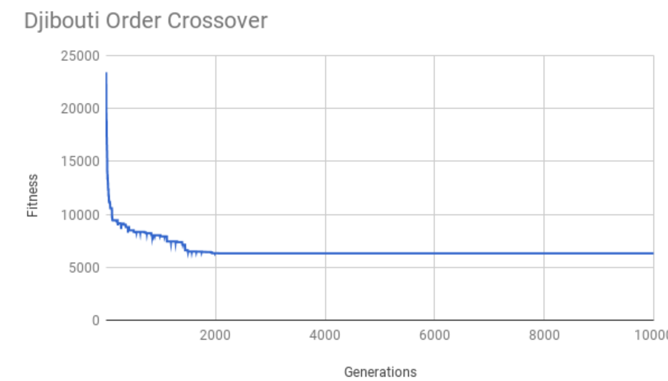
Djibouti38 Order Crossover	
Crossover Rate used	0.9
Mutation Rate used	0
Population Size	38
Seed used	55
Max generation	10000
Best FItness	15123.09153



Consistent with all of the other experiments conducted with the same parameters. Yields a very inefficient solution.

Djibouti38 Order Crossover d)

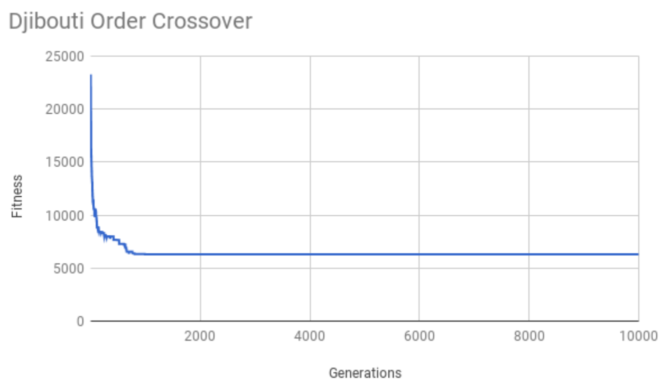
Djibouti38 Order Crossover	
Crossover Rate used	0.9
Mutation Rate used	0.1
Population Size	38
Seed used	55
Max generation	10000
Best FItness	6315.568273



Yielding a better solution than graph b), this may be indication Order Crossover benefits greater from crossover rates slightly less than 100 percent. Although a better fitness value is observed, a similar effect on the chromosome are observed where there is continual improvements in offspring until the 2000th generation. The plateau occurs and never seems to fluctuate.

Djibouti38 Order Crossover e)

Djibouti38 Order Crossover	
Crossover Rate used	0.9
Mutation Rate used	0.3
Population Size	38
Seed used	55
Max generation	10000
Best FItness	6300.17548



With the observations made on Djibouti38 dataset experiments on order crossover, an approach of crossover value slightly less than 100 percent is used, with 0.3 mutation rate. This yields a better solution than graph d) with similar effect on the offspring chromosomes plateau. This may be indication order crossover benefits from slightly higher mutation rates and slightly lower crossover rates.

II. Conclusion

It is observed that throughout the experiments held with both datasets, Order Crossover Yielded slightly better best fitness chromosomes. It can be seen that higher values of crossover rates and some mutation is more beneficial to Uniform Order Crossover than Order Crossover. While Crossover benefits more from slightly less crossover rate than 100 percent, and higher mutation rates than uniform order crossover. GA parameters impact the final outcome of chromosomes significantly with even a slight change. From the observations in the conducted experiments, Order Crossover is slightly more efficient than Uniform Order Crossover with the travelling Salesman problem. In conclusion, genetic algorithms proves to be efficient in determining optimal solutions to the Travelling Salesman Problem.

Acknowledgment

Dr. Ombuki-Berman for the relevant lectures and lecture material.
TA Jay Douglas for marking
TA Kyle Robert Harrison for the informative tutorials.

References

[1] M. Mitchell , An introduction to Genetic Algorithms
[2] Michalewicz, Genetic Algorithms + Data Structures = Evolution programs