



Genetic Algorithms Homework 2

By: Brian Ascencio,
Joel Membribe,
Stephen Maldonado

Representation 1 Description

Representation 1 was made to try to be able to run on our original Genetic Algorithm which meant we needed a unique way to represent the city order and calculate the fitness:

- Assigns city numbers to the lowest valued genes first (for duplicates, the first one that appears)
- Allows us to use Single Point Crossover and flip bit mutation as duplicates are allowed and there's no min/max number we can use

Pop Size	250	Crossover Rate	0.25
Num Gens	250	Mutation Rate	0.001
Selection Type	Tournament		
Scaling Type	Rank For Minimization		
Mutation Type	Flip Bit		
Crossover Type	Single Point Crossover		

Chromosome Genes

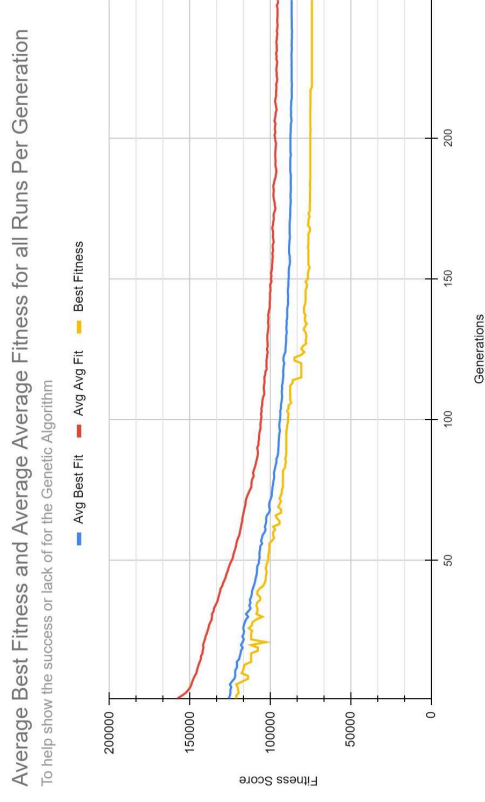
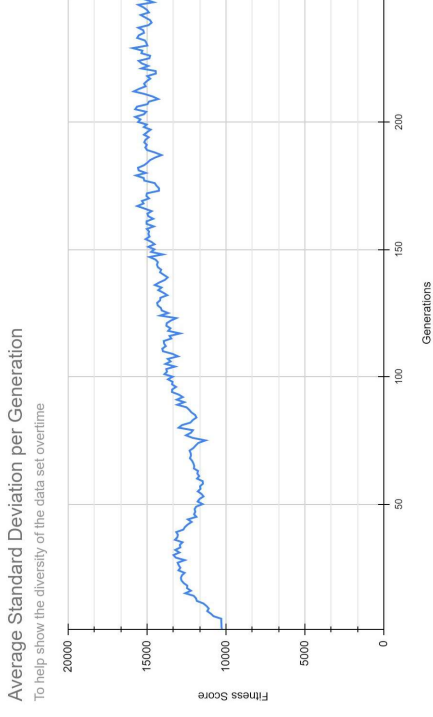
5	44	10	44	7
---	----	----	----	---

Respective City Order

1	4	3	5	2
---	---	---	---	---

Representation 1 Results

- Graphs (ATT48 dataset)
 - The top Chart shows that there is a increase in diversity for the first 50 generations, then it levels off.
 - The bottom Chart shows the GA representation does improve over time, but plateaus before reaching a “good” fitness
- Was unable to generate a “good” fitness
 - Best Fitness over 10 runs was ~73k, (6.8x the best known result)
 - Does show improvement from the randomly generated ~120k best fitness (11.2x the best known result)
- Similar with Berlin52 set and Best Fitness
 - Able to get from 3.3x the best known distance down to 2.5x the best known distance.



Representation 1 Discussion

Issues

- Disconnect between the ordering in the chromosome and the city visiting order
 - Cross Over does not exactly carry over the ordering
 - Mutation can cause a complete change in the city order

Pros

- Handled duplicates easily
 - Allowed single point crossover

Potential Improvements

- Implement elitism
 - Would help keep the better functioning paths

Extra Notes

- Tried implementing a similar design where the chromosomes' genes were overwritten by the city order they generated during the fitness calculation
 - Was hoping to improve crossover by limiting the range and preventing additional duplicates in the future
 - Performance was worse

Representation 2 Description

- Representation 2: selected a straightforward representation but modified genetic operation to see what path the representative took.
- Implemented to examine the results using the coordinates and calculate it using the Euclidean distance.
- Population size,mutation, crossover were mostly were adjusted to see the effects of uniform crossover.
- Listed all the cities and randomized each of them to do another type of crossover.
 - Enable us to use uniform crossover and then swap mutation

order	1	2	3	4	5
City	66	440	943	1201	446

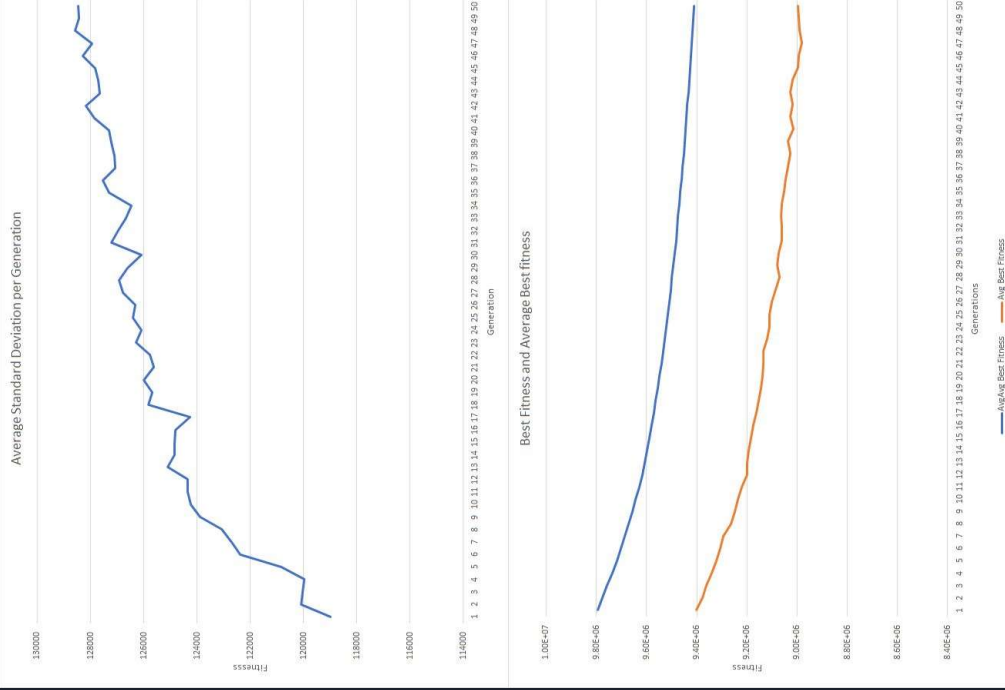
Generations	50
Population size	1050
Selection	Proportional
Scaling Type	Scale Minimization
Crossover Type	Uniform Crossover
Crossover Rate	1
Gene	1323
Mutation	Swap
Mutation Rate	0.0011

Representation 2 Results

Data

- RI1323 information as shown in the figure
- Increased for all 50 runs
- As we have more generations, graph levels off but not close to best known solution

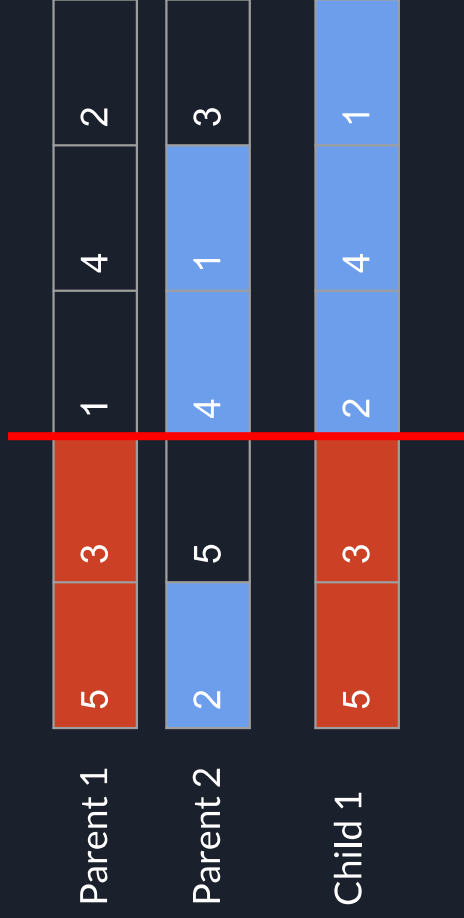
- Best known fitness wasn't generated but shows improvement as the generations increase.
- Best known solution
 - r1323 (880k)
 - Berlin52 (15k)
 - Att48 (95k)



Representation 3 Description

- We represented genes as cities, and chromosomes as a sequence of cities
- The order of the genes represented the path that the salesperson took
- Ordered single-point crossover and swap mutation were implemented

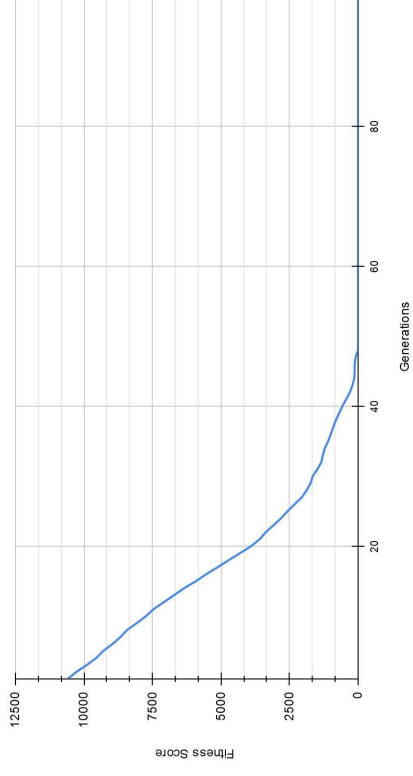
Pop Size	1000	Crossover Rate	0.5
Num Gens	100	Mutation Rate	0.01
Selection Type	Tournament		
Scaling Type	Rank For Minimization		
Mutation Type	Swap		
Crossover Type	Ordered Single Point Crossover		



Representation 3 Results

- Graphs shown are of att48.tsp
 - Best fit is found very quickly and rarely improves
 - Average fitness is varied but quickly converges, as shown by the avg avg fit line and the standard deviation
 - Converges quickly and does not explore the search space sufficiently
- Results were subpar; unsure as to exactly why
 - Perhaps, even with ordered single-point crossover, there was not enough exploration and variety
 - Uniform crossover may improve these results
 - Very poor fitness also seems to indicate a buggy implementation
- Similar results with other test files
 - Berlin52.tsp was around 230k
 - RL1323.tsp was around 9 million!

Average Standard Deviation per Generation
To help show the diversity of the data set overtime



Average Best Fitness and Average Average Fitness for all Runs Per Generation
To help show the success or lack of for the Genetic Algorithm

