Evolutionary computation

Genetic algorithms can help solving optimization problems, using the idea of evolution and survival of the fittest.

The schemata theory sees a genome in a string, where order is the number of fixed positions and length is the longest distance between them. The building block theorem suggests there are building blocks of low order and short length with higher probability for survival, and genetic algorithms recombine them for solution.

This have a impact on design. Genotype design should include as independent genes as possible, and related genes close together. Crossover probability should be low for less destruction.

The experiment tries to explore this design implications in two optimization problems – knapsack and salesman.

Knapsack problem

Given a bag and a set of n items, each with a weight and a value, find which items to include, to gain the maximum value in total, without exceeding the bag limit.

Here, the value corresponds to the item index, and the weight is a random number between 1-n. For comparison reasons, we fixate the problem as follows: population size = 10, generations = 50, n items = 50, bag limit = 20.

Design choices

The genotype represents a bag, as a list of all possible items, i.e. genes. Each gene is an item, and hold a Boolean value (1 if the item in the bag, or 0 if not).

- Selection- proportional (roulette wheel)
- Crossover – uniform.
- Mutation – include item or not.

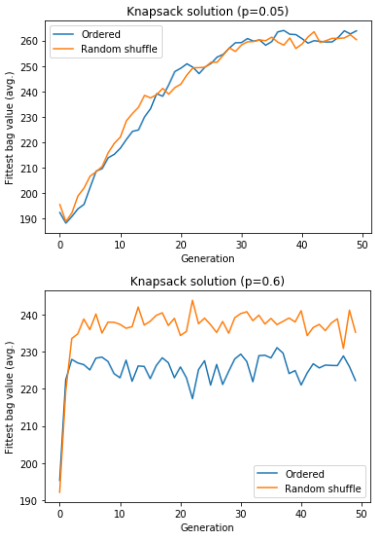
Independent variables:

- Genotype: the genes are independent, since each represent different item. However, we took different order of the genotypes: (1) ordered – genes are organized in the genotype according to the item index. Meaning, related genes (items with close values) are close to each other. (2) random order.
- Crossover and mutation: low (p=0.05) or high (p=0.6) probability for each.

Dependent variable

Best solution - bag with highest value.

Results: The average over 100 simulations:



Traveling salesman problem

Given a list of cities and their distances, find the shortest route that visits them all and returns to the staring point.

Here, the cities lie equidistantly on a unit circle. Therefore, the optimal solution (shortest path distance) of 360. For comparison reasons, we fixate the problem as follows: population size = 10, generations = 50, n cities = 6.

Design choices

The genotype represents a path, i.e. order of visiting the cities. Each gene is a city, and has no much information by itself. The information (and therefore diversity) lies in the order of them.

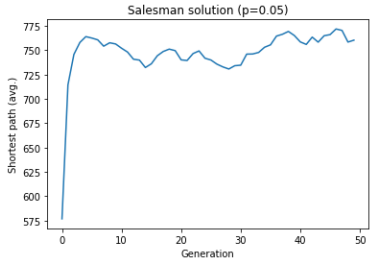
- Selection- proportional (roulette wheel)
- Crossover – CX2 (Hussain et al., 2017).
- Mutation – shuffling the cities that determined to be mutated in the path.

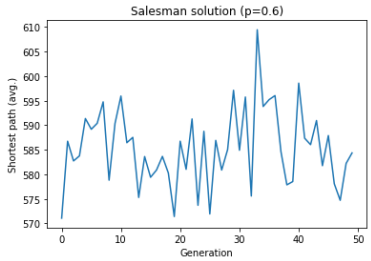
Independent variables:

- Genotype: as mentioned, the genes are highly dependent, but it is inherent. Also regarding locating similar genes close together, it is hard to do so without solving subproblems (since similarity here is by location, so if close cities will be close genes than it is part of the solution). Therefore, it did not act as an independent variable in this problem.
- Crossover and mutation: low or high probabilities were compared.

Dependent variable

Best solution - path with shortest distance.
Results: The average over 100 simulations:





Conclusion

It seems that low probability for building block destroy (crossover and mutation), indeed lead to better overall results, although more slowly. However, the ordered genotype (similar genes are closer) did not appear to improve the results. It might be due to limited similarity in the weights (which are random).

When the problem becomes more complex, such as the salesman problem which is NP-hard, the absence of independence between the genes in the genotype indeed showed limited evolution. As contrast to before, here the high probability lead to better although still limited results, and in the price of stability of the fittest.

References

Hussain, A., Muhammad, Y., Nauman Sajid, M., Hussain, I., Mohamd Shoukry, A., & Gani, S. (2017). Genetic Algorithm for Traveling Salesman Problem with Modified Cycle Crossover Operator. *Computational Intelligence And Neuroscience*, *2017*, 1-7. doi: 10.1155/2017/7430125