

Özyeğin University
CS551 – Introduction to Artificial Intelligence
Assignment-2

Solving Travelling Salesman Problem with Genetic Algorithm

Furkan Cantürk

04.30.2021

1. INTRODUCTION

In this assignment, Travelling Salesman Problem (TSP) is addressed by using a Genetic Algorithm (GA). For a given a symmetric or asymmetric cost matrix of n cities, a salesman is responsible to visit all cities and to return the depot. There are $(n-1)!$ tours for the asymmetric problem and $(n-1)!/2$ for the symmetric one. In this assignment, a given TSP of 20 cities with a symmetric cost matrix is solved by a design of a simple GA and tuning its parameters.

2. ALGORITHM

Genetic Algorithms (GAs) are a population-based heuristic solution procedure inspired by biological progression. GAs use chromosomes to encode solutions to an optimization problem. Each gene in the chromosome represents an attribute of that solution. Chromosome quality is evaluated by a defined fitness function, which is closely related to the objective function of the problem. A set of chromosomes is called generation and each generation is used to produce a new generation. Pairs of chromosomes are selected as parents based on fitness values to produce new chromosomes by using a crossover operator. Mutations are used to enrich population diversity to facilitate a thorough search of the feasible region. GAs maintains the common sections of the chromosomes that have high fitness values in populations throughout evolution. Although the general principles of GAs are common to all GA applications, there is no generic GA, and the user has to custom-design the algorithm for the problem at hand. In this assignment, some basic strategies for each method of GAs are utilized and each of them defined as follows.

Encoding: Permutation string of cities is used to represent the travelling order, e.g., 2-3-1-4-5 is one solution if there are 5 cities.

Fitness function: Reciprocal of total travelled distance.

Population size (n): Constant throughout the evolution and experimentally determined.

Initializing the population: Randomly generated n solutions.

Selection of parents: Parents to produce a new generation are selected by Tournament Selection (TS). n tournaments are done and the best chromosome among randomly selected (with replacement) k chromosomes is put to the mating pool in each tournament.

Crossover: Two parents at random are taken out of the pool to generate two sibling chromosomes. To achieve a feasible chromosome, which does not consist multiple genes (cities), an ordered crossover procedure is applied to parents. A section at random length of the first parent is randomly selected and then the remainder of the chromosome is filled with the genes from the second parent in the order in which they appear without duplicating any genes in the selected subset from the first parent. Ordered crossover is applied two times to given parents so that two new chromosomes are produced.

Mutation: Randomly two genes of each chromosome in the new generation is swapped with a probability of r , which is experimentally determined.

Selection of new population: If elitism approach is adopted, the current generation and the new generation is merged and n chromosomes which have the highest fitness value a are moved to next iteration of the evolution. Otherwise, only members of the new generation continue.

Number of generations (g): Number of iterations in evolution and experimentally determined.

3. EXPERIMENTAL EVALUATION

There are four parameters indicated in Section 4, which are population size (n), tournament size (k), mutation rate (r), usage of elitism (e), and number of generations (g). The following values of each parameter are tested, which combines up to 960 different parameter configurations:

$n \in \{25, 50, 100, 150, 200\}$, $k \in \{2, 3, 4, 5\}$, $r \in \{0, 0.2, 0.4, 0.6, 0.8, 1\}$, $e \in \{0, 1\}$, and $g \in \{25, 50, 100, 200\}$.

To provide a sound tuning of parameter configurations, the same initial population is used for all of them. Also, 5 runs are executed with different random seeds for each configuration since GAs are stochastic algorithms which might yield different results under different random seeds.

The following table presents distance values of the best solution at the last generation for each random seed of each parameter configuration. These 5 distance results are averaged at the last column of Table 1. Configuration-1 (C-1) consists the best solution with a distance value of 871.1 (Sol-1) at each run. Compared to C-1, the next 7 configurations achieve a close performance and each of 7 provides Sol-1 4 times out of 5 runs. The best solution of the initial population has a distance value of 2560, which indicates the effectiveness of the designed GA for the given TSP instance. However, it does not represent a general performance for TSP since it is tested using only one problem instance.

These best 15 configurations shows that elitism rule applied to select the next population provides better performance. Elitism approach utilized in this assignment would reduce the diversity of solutions and would cause to converge to local maximum points earlier, but it seems that it is not the case for the given problem, which is not a high dimensional TSP (only 20 cities).

For the result analysis of other parameters according to Table 1, population size should be more than 100, mutation rate should be around 0.4, and number of generations should be more than 100. Lastly, it seems that effect of tournament size was not significant since it varies in each configuration in the table.

Config.	e	n	r	s	g	Run-1	Run-2	Run-3	Run-4	Run-5	Avg.
1	1	150	0.4	2	200	871.1	871.1	871.1	871.1	871.1	871.1
2	1	100	0.2	4	200	871.1	890.2	871.1	871.1	871.1	874.9
3	1	100	0.6	4	200	871.1	871.1	871.1	890.2	871.1	874.9
4	1	200	0.0	2	100	890.2	871.1	871.1	871.1	871.1	874.9
5	1	200	0.0	2	200	890.2	871.1	871.1	871.1	871.1	874.9
6	1	200	0.2	2	200	871.1	871.1	890.2	871.1	871.1	874.9
7	1	200	0.4	2	100	871.1	871.1	890.2	871.1	871.1	874.9
8	1	200	0.4	2	200	871.1	871.1	890.2	871.1	871.1	874.9
9	1	200	0.6	4	100	871.1	871.1	871.1	897.5	871.1	876.4
10	1	200	0.6	4	200	871.1	871.1	871.1	897.5	871.1	876.4
11	1	100	0.2	3	200	871.1	897.5	871.1	871.1	871.1	876.4
12	1	200	0.2	2	100	871.1	880.8	890.2	871.1	871.1	876.9
13	1	200	1.0	2	200	871.1	890.2	880.8	871.1	871.1	876.9

14	1	150	0.2	5	100	890.2	871.1	871.1	871.1	890.2	878.8
15	1	150	0.2	5	200	890.2	871.1	871.1	871.1	890.2	878.8

Table 1. The best 15 parameter configurations