FOLIA 180

Annales Universitatis Paedagogicae Cracoviensis

Studia Technica VIII (2015)

Wiktor Hudy, Kazimierz Jaracz

Influence of the amount of generations and the likelihood of mutations on the results of an evolutionary algorithm adapted to solve the travelling salesman problem

Introduction

Travelling Salesman Problem involves generating points in the simplest case, on the plane. The salesman is intended to visit all the points generated exactly once and return to the point from which took off. The length of the path travelled is the criterion of quality. The shorter the road, the better. The problem is to find the shortest path that travelling salesman has to go though [1, 2, 3, 5].

This task is easy to formulate. Solving this is no longer so simple. Your best bet would be to search all the possible roads of a Salesman. The number of roads is a permutation of the points [1, 5], Table 1 shows the number of possible roads for the number of points example.

Tab. 1. Summary of the number of points and the corresponding with them an amount of roads that the travelling salesman would travel

Number of points	Number of possible roads
5	120
10	362880
20	$2.43 \cdot 10^{18}$
50	$3.04 \cdot 10^{64}$
100	$9.33 \cdot 10^{157}$

It is possible to search all roads for about ten points in the plane. Even with a few dozen or more points the search of all possible roads becomes unprofitable due to the computation time. A case discussed is the simplest possible, because there are basic modifications to the travelling salesman problem, such as:

- roads between the points are not distances from these points (then the table of roads' costs is introduced as a replacement),
- points are not on the plane,
- roads between the two points are different depending on the march direction (from point A to B one is a different value of the fitness function than from point B to A). In order to find this not always optimal path, an evolutionary algorithm is used.

Evolutionary Algorithm

The general scheme of evolutionary algorithm is shown in Figure 1.

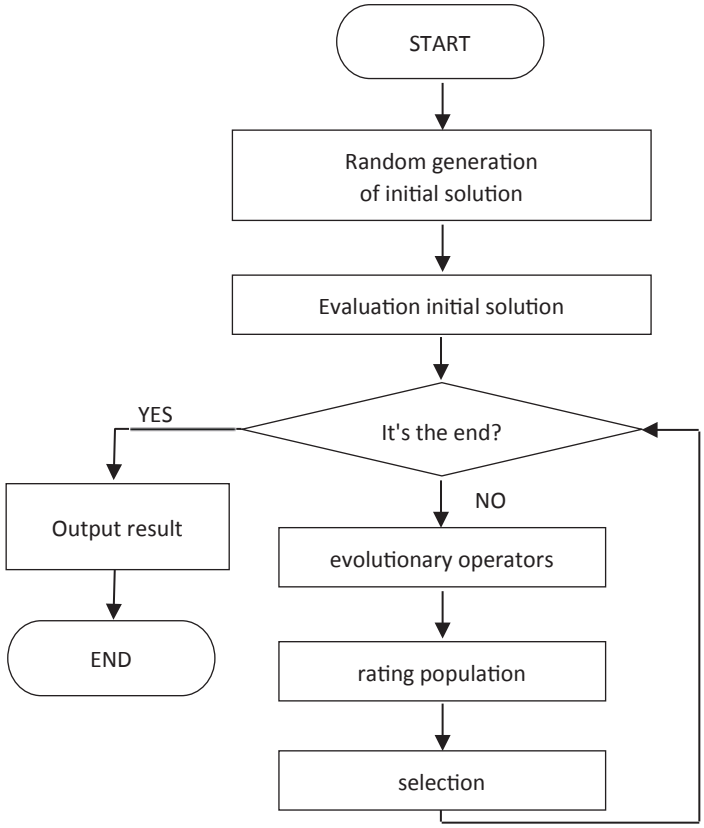

Fig. 1. A block diagram of a genetic/evolutionary algorithm [3, 4]

After the start of the algorithm, initial population is generated, which consists of randomly generated individuals. Then these individuals are evaluated, which means that the value of the fitness function is assigned. The value of this function determines how good a given individual is. In the case of minimizing the quality indicator, the smaller value of the evaluation function has the individual, the better he is (or otherwise, is better suited). The process of evolution continues inside the close loop. Genetic operators create new individuals on the basis of the current population of individuals. Then individuals are evaluated. The selection operator selects individuals from the current population, which is enlarged for individuals generated by the genetic operators to the new population.

New population becomes the current one in the next program loop. After the fulfilment of the condition which ends the process of evolution, the result of the calculation is output [2, 3, 4].

Studies of the algorithm for solving the TSP

The population size was set a'priori and set at 50 individuals. Similarly, the number of points in the plane has been set at 50. The following were examined:
- number of generations (parameter directly responsible for the duration),
- mutation probability.

A series of tests for the same points on the plane was done. The study concentrated on the point where changes were observed. The results are presented in Table 2. The value of the fitness function (F) are expressed in relative units.

Tab. 2. Dependence of the probability (pm) of the number of generations (P)

Number of generations P	Value of mutation probability (p_m)								
	0.01	**0.02**	**0.03**	**0.04**	**0.05**	**0.08**	**0.1**	**0.2**	**0.5**
10	15006.0	15905.6	15541.2	15693.6	15485.3	17004.6	15701.8	16681.1	15918.6
20	15602.7	14090.5	14071.3	14136.4	14016.7	13899.5	12928.9	12689.0	14986.7
50	12996.4	11378.6	11734.5	13008.8	12365.2	12716.8	12380.2	13156.7	13915.1
80	10446.5	11534.5	11831.6	10751.1	12289.8	12678.9	10595.4	11883.4	12482.1
100	11011.2	10642.0	11956.7	10386.8	12105.6	11080.6	10721.5	10488.8	13801.5
200	9785.87	10363.2	10056.4	10755.3	9100.08	10245.7	11054.1	10243.6	11312.7
500	9122.95	8488.04	8995.94	9075.00	10033.9	10216.1	10309.0	11049.8	11853.7
800	9477.38	7848.11	9211.37	8722.34	8721.79	9373.09	9266.93	10710.4	11526.2
1000	8285.97	8267.69	7763.74	9429.99	9492.16	8419.73	10071.2	10238.0	11563.8

Results are presented graphically in Figure 2.

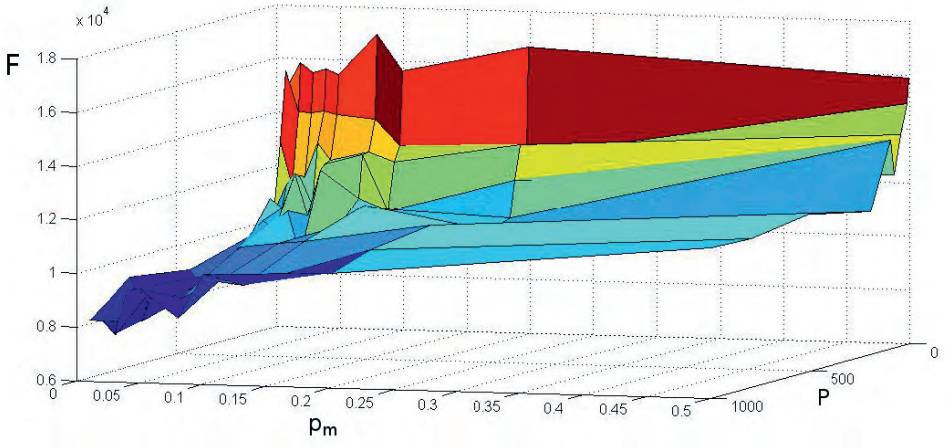

Fig. 2. Figure quality indicator (F) depending on the probability of mutation (pm) and on the number of generations (P)

Figure 3 shows a graph of time required for the calculation of the number of generations. All the results were averaged for the calculation time.

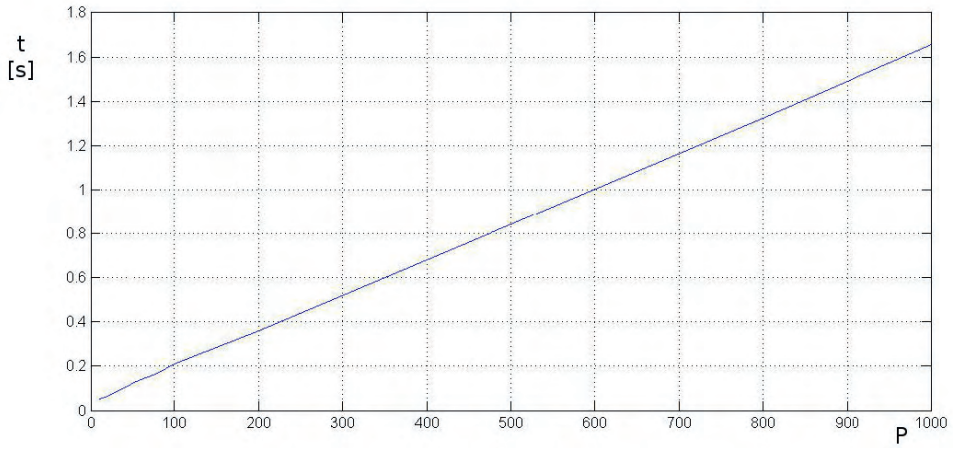

Fig. 3. Dependence of the time required to make calculations by the evolutionary algorithm on the number of generations

Figure 4 illustrates the shortest found path corresponding to the values in Table 2 pm = 0.02, P = 800.

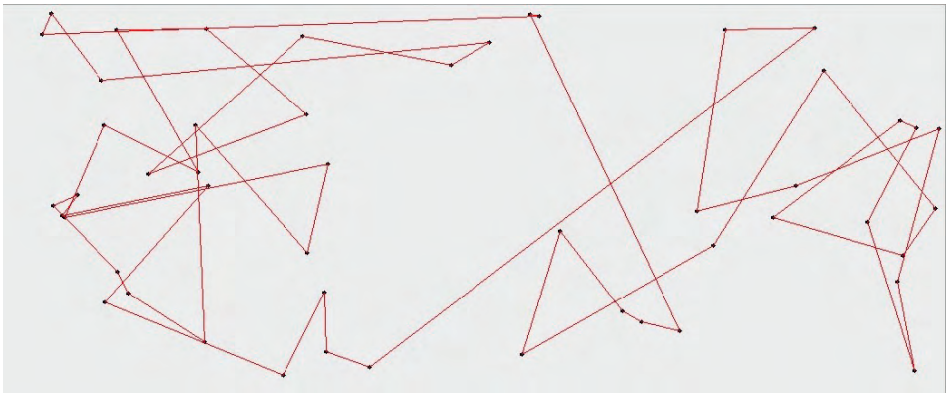

Fig. 4. The shortest found path with marked points through which a travelling salesman passed

Conclusion

As the calculations show, the result is affected by the number of generations and the value of rate of the mutation probability. The greater the number of generations, the shorter road the algorithm found. Unfortunately, then the computing time grew linearly (Fig. 3). The effect of rate of the mutation probability was not linear. It had a number of local minima (Fig. 2). The best result was obtained for values of pm = 0.02. The effect of the number of generations was also not linear and resembles

exponential function (Fig. 2). Further increase of the number of generations will not affect the outcome significantly. As can be seen from Figure 4, the road found is not the shortest one, probably the elongation of algorithm work time could improve the result. The result obtained is suitable as input stage to the other optimization algorithms.

References

[1] Goldberg D.E., *Algorytmy genetyczne i ich zastosowania*, WNT, Warszawa 1998.
[2] Gwiazda T.D., *Algorytmy genetyczne. Kompendium*. T. 1–2, PWN, Warszawa 2007.
[3] Hudy W., Jaracz K., *Influence of evolutionary algorithm's parameters on optimization quality, on a basis of travelling salesman problem (Wpływ parametrów algorytmu ewolucyjnego na jakość optymalizacji na przykładzie problemu komiwojażera)*. Annales Academiae Paedagogicae Cracoviensis. Studia Technica IV. 2011, 120(6), s. 48–53.
[4] Hudy W., Jaracz K., *Analysis of the influence of mutation operation in evolutionary optimization parametric method of field-oriented control system with induction low-power engine (Analiza wpływu operatora mutacji postępowej w ewolucyjnej metodzie optymalizacji parametrycznej układu sterowania polowo-zorientowanego z silnikiem indukcyjnym małej mocy)*. XIX ZKwE '2014, Poznań 2014, April 28–29.
[5] Michalewicz Z., *"Algorytmy genetyczne + struktury danych = programy ewolucyjne"*, WNT, Warszawa 1999.

Abstract

The article examined the influence of the amount of generations and the impact of the mutation probability of the output of the evolutionary algorithm. Evolutionary algorithm was designed to solve the travelling salesman problem. Other parameters, including initial population size and the number of points, have been set a'priori. Tested points were generated randomly.

Key words: evolutionary algorithm, Travelling Salesman Problem

Wiktor Hudy, Kazimierz Jaracz
Pedagogical University of Cracow
Institute of Technology
ul. Podchorążych 2
30-084 Kraków, Poland