Genetic Algorithm for Solution of the Traveling Salesman Problem with New Features against Premature Convergence

Article · December 1996

Source: CiteSeer

3 authors, including:

Victor V Miagkikh
Institute of Electrical and Electronics Engineers

Genetic Algorithm for Solution of the Traveling Salesman Problem with New Features against Premature Convergence

Victor M. Kureichick, Victor V. Miagkikh, Alexander P. Topchy

Taganrog State University of Radio-Engineering
Nekrasovsky L. 44, GSP-17A, Taganrog, 347928, RUSSIA
e-mail: kur@itl.rnd.su

ABSTRACT

Genetic Algorithms are finding increasing number of applications in a variety of problems in a whole spectrum of disciplines. This paper describes a genetic algorithm with novel features against premature convergence successfully applied to solution of the Traveling Salesman Problem (TSP). Authors practiced modified version of Greedy Crossover, and operators brushing the population helping to escape from local minima. As the results, the frequency of finding the optimal solution was improved with keeping good convergence. Developed application was tested on a number of benchmarks: Oliver's 30, Eilon's 50 and Eilon's 75 towns TSPs. In 30 towns problem the algorithm seems to be always reaching an optimal solution. In 50 and 75 towns new better tours were found in comparison to all ones available to authors via papers on similar investigations. Proposed approach in TSP solution is generalized as *the social disasters technique.*

INTRODUCTION

The TSP is defined as a task of finding of the shortest Hamiltonian cycle or path in complete graph of *n* nodes. This well known problem has such an interpretation: A salesperson should visit a set of towns in such a way, that to attend all city only once and return to the starting city spending the least possible sum of money. The towns are denoted by nodes in the weighted graph. The weighted on the edges of this graph defines the cost of transportation. TSP is a classic example of an NP-hard problem. The methods of finding of the optimal solution involve search in a solution space that grows exponentially with number of towns. Therefore, heuristic techniques are applied to reduce the search space and direct it to the areas with the highest feasibility of good solutions. Ones of such heuristics are Genetic Algorithms (GAs).

GAs are based on the ideas of popular genetics. It is an iterative procedure which maintains a population of candidate solutions. These solutions (instances or chromosomes) are encoded into strings of symbols from a fixed alphabet. The initial population of instances, represented by their chromosomes, can be chosen heuristically or at random. During each iteration step, called a generation, under a number of selected from population solutions idealized genetic operators are performed. Two major genetic operators are crossover and mutation. Crossover operator on the basis of parental instances produces offspring(s) solution, which inherits some features from parents. F.e., if the chromosomes are represented as binary strings, then crossover can be implemented by choosing a point at random, called the crossover point, and exchanging the parts of structures located to the left/right from it. Let p1=010│110101 and p2=100│10010, and suppose that the crossover point has been chosen as shown by│. The resulting structures are o1=01010010 and o2=100110101. Mutation operator makes random changes in chromosomes introducing new search areas. New population is formed from obtained new instances as well as the part of the former population.

There were a lot of more or less successful attempts to apply GAs to TSP solution, i.e. [1], [2], [3], [6] and their references. The chromosome is usually coded as a succession of visited towns or as a list of edges in the candidate Hamiltonian path. The restriction in the definition of the problem, that there should not be towns twice or not visited, cause ordering dependencies in representation and require specially constructed crossovers. Such crossovers for TSP may be found in [2] - Cycle Crossover, Partially-Mapped Crossover - PMX [5], Grefenstette or Greedy Crossover [1]. The Grefenstette crossover is defined by the following rule:

1. For each pair of parents, pick a random city for the start.

.

2. *Compare the two edges leaving the city (as represented in two parents) and choose the shorter edge.*
3. *If the shorter parental edge would introduce cycle into the partial tour, then extend the tour by a random edge.*
4. *Continue to extend the partial tour using steps two and three until the circuit is complete.*

Grefenstette crossover is called 'greedy' since it always prefers locally cheaper paths. It is based on the knowledge about the problem, that locally cheaper choice in most cases is better then more expensive one. Of course, such a simple rule may not always drive to optimum, because sometimes it is necessary to sacrifice instant profits for the sake of the global savings. Nevertheless, crossovers based on knowledge improve convergence, but they may drive to local minima, which GA with usual measures against premature convergence are not able to get out.

Premature convergence is one of the major difficulties with GA and in fact with most search algorithms. It has been observed that this problem is closely tied to the problem of losing diversity in the population. One source of it is occasional appearance of 'super-individual(s)' which in few generations takes over the population. There are proposed many ways for avoiding this problem [5]. However, in case of greedy genetic [7] the situation with premature convergence seems to be worse.

Analyzing the behavior of standard greedy crossover for TSP authors have seen the picture reported many times - the loss in diversity, observed in abundance of the same tours varied in a starting town. Authors directed efforts to elimination of the following two reasons of this situation, which were considered to be the major ones: *1. Immoderate crossover greediness; 2. Low influence of random factors intended to make the population get out of local minima.*

MODIFICATIONS IN GREEDY CROSSOVER

Standard Greedy Crossover very easily kills all 'bad' changes which are produced by randomizing genetic operators. It was experimentally noted, that recombination of the best ranking individual with not so good one almost always produces an offspring, which is the same as the better of the parents. If we place such an individual into the population, then it will not introduce anything new. Such 'good' solutions will result in inefficient usage of the population pool. There are some ways to avoid it: change the crossover or don't place the offspring. The following measures in crossover may be undertaken. It is acceptance of more expensive path to the cheaper with a definite probability or taking about fifty-fifty of the parental representations. Random picking of a town in case of deadlock or choice of the closest may be optional. The latter may be useful as an additional heuristic, because it is known [1], that transferring edges from parents to offspring is only 60% effective, which implies that the left 40% are chosen randomly. The fifty-fifty approach results in the following crossover rule, which is oriented to application with TSP defined upon the undirected graphs: *To find a random town in the first parent, that becomes the current town. Current parent is the first parent. Body of the crossover: decide, which of two adjacent towns ('adjacent towns' means located in the positions to the left and to the right from the current town in chromosome) in the current parent is closer to the current town. The closer, if not visited, becomes the current town. If only one of the adjacent towns is not visited then it will become the current town. If both adjacent towns are visited, then we'll need to choose the closest from the number of remaining unattended. Current parent is another parent. Repeat the body until all the towns are visited. Offspring is forming from the succession of the towns in order they were becoming current towns.*

Greedy Crossover with variable probability of bad solutions acceptance requires tuning of this probability, if it is a constant, or special function for its automatic variation in run time. Authors have not implemented this case. All the results were obtained on the fifty-fifty modification of the Greedy crossover.

OPERATORS WORKING WITH WHOLE POPULATION.

Decrease of the crossover greediness will be effective against the lack of variety in the population only if there is good income of new changes, produced by random factors or additional heuristics. After reaching deep local minimum, there are usually a lot of the same tours. Authors decided to construct special operators for striving against it directly. So, after determining of being in local minimum, the special algorithm, changing a number of individual in the population, is activated. The fact of reaching local (global) minimum in TSP case is concluded when N-first individuals have the same length of tour. We called this N an *elite number*. We assume, that in TSPs with a large number of towns, the equal length indicates the same tour configuration. This rule works perfectly for the distances of the real type, and practice shown it to be valid in case of integer values. There are some other ways to determine this, like not improvement of the best tour's length after K trails. When suboptimum is determined one of the following operators may be performed:

1. **Packing.** *Of all the tours, having the same length, only one remains unchanged. Under all the others the Warp Operators are performed.*
2. **Judgment_Day.** *Only the best instance remains unchanged. Under all the others the Warp Operators are performed.*
   Warp Operators are full or partial randomization.

Proposed operators are similar to social disasters in real population. They may seem to be extremely severe, but the analysis of the evolution after execution of this operators shows, that the influence of the best ranking instance is very strong. After about $5 \cdot Population\_Size$ performed crossovers (after full-except-first randomization) in case of proposed modified greedy crossover, population returns to the very good form and soon even new activation may be needed. Comparing Packing and Judgment_Day authors noted almost the same effect, and Judgment_Day operator may be preferred, because of its simplicity.

EXPERIMENTAL RESULTS

In our application the Roulette Wheel selection and the replacement strategy picking up the worst were implemented. Distance matrix calculation procedure and evaluation function are taken from [3]. Fifty-fifty greedy crossover and Judgment_Day operator were used. Own object oriented GUI environment, working under MS Windows was developed. It is free software and available on the request via e-mail.

The application was tested on a number of benchmarks: Oliver's 30 [2], Eilon's 50 and Eilon's 75 (both taken from [3]) towns TSPs. Thirty towns problem was solved the most easily. Being good tuned (Population_Size=20, mutation_rate = 0.3, elite_number=10), the GA reached the optimum in every run. In average (100 runs) GA reaches optimum solution after about 5000 genetic operators performed (see fig. 1). 50 and 75 towns TSPs seem to be harder for this GA. There are still many local minima which it strives with difficulties. The tours on fig. 2 and fig. 3 were located in about a half of all runs. But in comparison with ones in [3] this tours are shorter (the full list of towns' coordinates and their sequence is given in the Appendix).

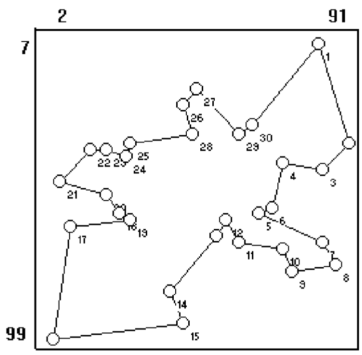

Fig. 1. Best obtained tour in Oliver's 30 towns TSP (Length=419).

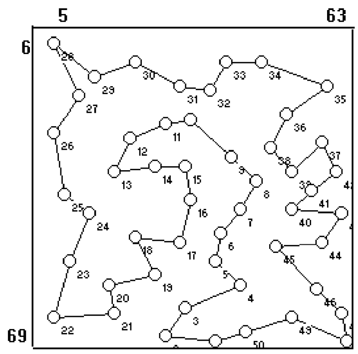

Fig. 2. Best obtained tour in Eilon's 50 towns TSP (Length=425).

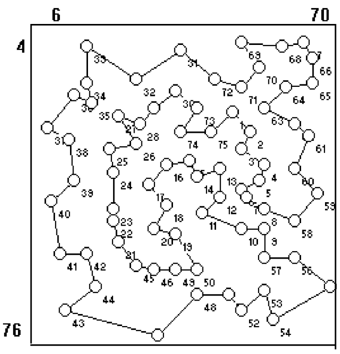

Fig. 3. Best obtained tour in Eilon's 75 towns TSP (Length=535).

DISCUSSION AND CONCLUSIONS

According to TSP benchmark, the use of described method brings significant improvements. Authors consider its success as a consequence of the following factors: i) very efficient usage of the population pool,

.

due to the regular 'purges' produced by proposed operators. This statement is supported by the fact, that this GA is very efficient even with small sizes of the population (about 20) working with TSPs of large number of towns. ii) Powerful ‘pushes’ produced by Warp operators helps the population to get out of the local minima. iii) Modifications in crossovers guaranties, that random changes, produced by mutations and randomizing operators will not be immediately ‘corrected’. iiii) Good general performance is the result of both crossover modification and operators, working with whole population, because experiments turned out that one without another works not so well.

In general, proposed modifications highly influenced onto the ability to escape from local optima of the GA having the crossover based on local preferences. As the result of their application, the frequency of obtaining an optimal solution was improved with keeping good convergence. However, for more complicated TSPs the combination of proposed modification with local improvement operators [6] may be useful. It is also very interesting to try this features in multi-populational (parallel) GAs using various migration schemes.

Results in TSP encourage to try this approach in genetic solution of other problems in Computer Aided Design [4] and in different domains of Artificial Intelligence. Now we are applying this technique to the Graph Coloring Problem and the Quadratic Assignment Problem. Preliminary results shows, that two generalized points of the proposed approach called *social disasters technique* also brings improvements, being applied to these problems. These points are: i) *Greedy (local improvement) crossover with moderate greediness.* ii) *Randomizing operators working with whole population.* Results of application of this technique to these two problems will appear in a separate paper.

APPENDIX

The new best solution for Eilon's 50 city problem, giving coordinates in order of the cities visited:
(32,22),(27,23),(20,26),(17,33),(25,32),(31,32),(32,39),(30,48),(21,47), (25,55),(16,57),(17,63),(5,64), (8,52), (12,42),(7,38),(5,25),(10,17),(5, 6),(13,13),(21,10), (30,15),(36,16),(39,10),(46,10),(59,15), (51,21),(48,28), (52,33),(58,27),(61,33),(56,37),(52,41),(62,42),(58,48), (49,49),(57,58), (62,63),(63,69), (52,64), (43,67),(37,69), (27,68),(31,62),(42,57),(37,52),(38,46),(42,41),(45,35), (40,30).
Length = 425. The best tour length in [3] is 428.

The new best solution for Eilon's 75 city problem giving coordinates in order of the cities visited:
(30,20),(27,24),(22,22),(26,29),(20,30),(21,36),(21,45),(21,48),(22,53),(26,59),(30,60),(35,60),(40,60), (35,51),(30,50),(33,44),(29,39),(33,34),(38,33),(40,37),(45,35),(45,42),(41,46),(50,50),(55,50),(55,57), (62,57),(70,64), (57,72),(55,65),(50,70),(47,66),(40,66),(31,76),(10,70),(17,64),(15,56),(9,56),(7, 43), (12,38),(11,28),(6,25), (12,17), (16,19), (15,14),(15, 5),(26,13),(36, 6),(44,13), (50,15),(54,10), (50, 4), (59, 5),(64,4),(66, 8),(66,14), (60,15), (55,20),(62,24),(65,27), (62,35),(67,41),(62,48), (55,45),(51,42), (50,40),(54,38),(55,34),(50,30),(52,26), (48,21),(43,26),(36,26),(40,20),(35,16).
Length = 535. The best tour length in [3] is 545.

The following is the fragment of the function, calculating distances between the towns.

```
x=towns[i].x-towns[j].x;
y=towns[i].y-towns[j].y;
Paths[i][j]=int(sqrt( x*x+y*y)+0.5);
```

REFERENCES

[1] Grefenstette J *et al*, 1985, ’Genetic algorithms for the traveling salesman problem’, Proc. Intern. Conf. of Genetic Algorithms and their applications. Pp:160-165.

[2]Oliver I *et al*, 1987, ‘A study of permutation crossover operators on the traveling salesman problem’, Proc. of the Second Int. Conf. on Genetic Algorithms, Pp: 224-230.

[3] Whitley *et al*, 1989, ‘Scheduling Problems and Traveling Salesman: The Genetic Edge Recombination Operator’, Proc. of the Third Int. Conf. on Genetic Algorithms. Pp:133-140.

.

[4]Goodman E *et al*, 1994, 'A Genetic Algorithm Approach to Compaction, Bin Packing, and Nesting Problems', Tech. report #940702. Case center for computer-aided engineering and manufacturing. Michigan State University.

[5] Davis L, 1991, 'Handbook of genetic algorithms', Van Nostrand Reinhold. New York.

[6] Jog P *et al*, 1989, 'The Effect of Population Size, Heuristic Crossover and Local Improvement on a Genetic Algorithm for the Traveling Salesman Proble' Proc. of the Third International Conf. on Genetic Algorithms. Pp:110-114.

[7] Leipins G. *et al*, 1987, 'Greedy Genetics', Proc. of the Second International Conf. on Genetic Algorithms, Pp: 90-99.