

## Orthogonal Genetic Algorithm and Its Application in Traveling Salesman Problem

Hanmin Liu<sup>1, 3,a</sup>, Qinghua Wu<sup>2,b</sup> and Xuesong Yan<sup>3,c</sup>

<sup>1</sup>Wuhan Institute of Ship Building Technology, Wuhan, China

<sup>2</sup>School of Computer Science and Engineering, Wu-Han Institute of Technology, Wuhan, China

<sup>3</sup>School of Computer Science, China University of Geosciences, Wuhan, China

<sup>a</sup>yanxs1999@126.com, <sup>b</sup>wuqinghua@sina.com, <sup>c</sup>yanxs@cug.edu.cn

**Keywords:** orthogonal genetic algorithm, local search, fitness function

**Abstract.** The traveling salesman problem (TSP) is one of the most widely studied NP-hard combinatorial optimization problems. Its statement is deceptively simple, and yet it remains one of the most challenging problems and traditional genetic algorithm trapped into the local minimum easily for solving this problem. Therefore, based on a simple genetic algorithm and combine the base ideology of orthogonal test then applied it to the population initialization, crossover operator, as well as the introduction of adaptive orthogonal local search to prevent local convergence to form a new orthogonal genetic algorithm. The new algorithm shows great efficiency in solving TSP with the problem scale under 300 under the experiment results analyze.

### Introduction

The most popular evolutionary model used in the current research is Genetic Algorithms (GA), originally developed by John Holland [1]. The GA reproduction operators, such as recombination and mutation, are considered analogous to the biological process of mutation and crossover respectively in population genetics. The recombination operator is traditionally used as the primary search operator in GA while the mutation operator is considered to be a background operator, which is applied with a small probability. Traditionally, GA uses a binary string representation of chromosomes with concentration on the notion of 'schemata'. A schema is a template that allows exploring the similarity among chromosomes. Genetic Algorithms model evolution as a search for structures or building blocks that perform well in a given environment. Therefore, the recombination and mutation operators focus on an individual's structure, not the structure's interpretation. The results of applying reproduction operation in GA generate solutions that share structural similarities with their parents but may have significantly different interpretations. However, many recent applications of GA have used other representation such as graphs, Lisp expressions, ordered list, and red-valued vectors.

t=0;

initialize P(t)

evaluate structures in P(t)

repeat

    t= t+1

    select- reproduction C(t)from: P(t-1)

    recombine and mutate structures in C(t)forming C'(t);

    evaluate structures in C'(t)

    select-replace P(t) from C '(t) and P(t+ 1);

Until ( termination condition satisfied ).

Above gives the basic algorithmic steps for GA. After the initial population of individuals is generated (usually randomly) and individuals' structures are evaluated, the loop is entered. Then a selection buffer C(t) is created to accommodate the selected copies from P(t-1), "select-reproduction".

In the Holland original GA, individuals are selected probabilistically by assigning each individual a probability proportional to its structural fitness. Thus, better individuals are given more opportunity to produce offspring. Next the variation operators (mutation and crossover) are applied to the individuals in  $C(t)$  buffer producing offspring  $C9(t)$ . After evaluating the structural fitness of  $C7(t)$ , the selection method is applied to select replacement for  $P(t)$  from  $C'(t)$  and  $P(t-1)$ . In general, genetic algorithms are usually used to solve problems with little or no domain knowledge, NP-complete problems, and problems for which near optimum solution is sufficient. The EA methods can be applied only if there exist a reasonable time and space for evolution to take place.

The traveling salesman problem (TSP)[2] is one of the most widely studied NP-hard combinatorial optimization problems. Its statement is deceptively simple, and yet it remains one of the most challenging problems in Operational Research. The simple description of TSP is: Give a shortest path that covers all cities along. Let  $G = (V; E)$  be a graph where  $V$  is a set of vertices and  $E$  is a set of edges. Let  $C = (c_{ij})$  be a distance (or cost) matrix associated with  $E$ . The TSP requires determination of a minimum distance circuit (Hamiltonian circuit or cycle) passing through each vertex once and only once.  $C$  is said to satisfy the triangle inequality if and only if  $c_{ij} + c_{jk} \geq c_{ik}$  for all  $i, j, k \in V$ .

Due to its simple description and wide application in real practice such as Path Problem, Routing Problem and Distribution Problem, it has attracted researchers of various domains to work for its better solutions. Those traditional algorithms such as Cupidity Algorithm, Dynamic Programming Algorithm, are all facing the same obstacle, which is when the problem scale  $N$  reaches to a certain degree, the so-called "Combination Explosion" will occur. For example, if  $N=50$ , then it will take  $5 \times 10^{48}$  years under a super mainframe executing 100 million instructions per second to reach its approximate best solution.

A lot of algorithms have been proposed to solve TSP[3-8]. Some of them (based on dynamic programming or branch and bound methods) provide the global optimum solution. Other algorithms are heuristic ones, which are much faster, but they do not guarantee the optimal solutions. There are well known algorithms based on 2-opt or 3-opt change operators, Lin-Kernighan algorithm (variable change) as well algorithms based on greedy principles (nearest neighbor, spanning tree, etc). The TSP was also approached by various modern heuristic methods, like simulated annealing, evolutionary algorithms and tabu search, even neural networks.

### Orthogonal Genetic Algorithm for TSP

**Chromosome Coding:** In this paper, we will use the most direct way to denote TSP-path presentation. For example, path 4-2-1-3-4 can be denoted as (4, 2, 1, 3) or (2, 3, 1, 4) and it is referred as a chromosome. Every chromosome is regarded as a validate path. (In this paper, all paths should be considered as a ring, or closed path).

**Initialize population:** The traditional method of genetic algorithm is randomly initialized population, that is, generate a series of random numbers in the solution space of the question. In this paper, the orthogonal genetic algorithm uses the orthogonal initialization [9]. For the general condition, before seeking out the optimal solution the location of the global optimal solution is impossible to know. for some high-dimensional and multi-mode functions to optimize, the function itself has a lot of poles, and the global optimum location of the function is unknown. If the initial population of chromosomes can be evenly distributed in the feasible solution space, the algorithm can evenly search in the solution space for the global optimum. Orthogonal initialization is to use the orthogonal table has the dispersion and uniformity comparable, the individual will be initialized uniformly dispersed into the search space, so the orthogonal design method can be used to generate uniformly distributed initial population.

**Adaptive local search operator:** Local search operator has a strong local search ability, and then can solve the shortcomings of genetic algorithm has the weak ability for the local search. And the population according to the current state of adaptive evolution of the local search space adaptive local search operator will undoubtedly greatly enhance the ability of local search. In the initial stage of the

evolution, the current optimal solution from the global optimum region is still relatively far away, this time the adaptive local search operator to require search a large neighborhood space to find more optimal solution, it can maintain the population diversity. When the population has evolved to the region containing the global optimum, the adaptive local search operator to require a relatively small area to search in order to improve the accuracy of the global optimal solution.

In this paper, the adaptive local search operator is the adaptive orthogonal local search operator. Adaptive orthogonal local search operator is aimed at the neighborhood of a point to search, so the key point is to identify a point as the center of the hypercube, the hypercube in the orthogonal test, expect to be better Solution.

Fitness function: The only standard of judging whether an individual is “good” or not. We take the reciprocal of the length of each path as the fitness function. Length the shorter, fitness values the better. The fitness function is defined as following formula:

$$f(Si) = \frac{1}{\sum_{i=1}^N d[C_{n(i)}, C_{n(i+1) \bmod N}]} \quad (1)$$

Framework of algorithm:

Step 1: Orthogonal initialize population;

Step 2: Using selection method select the excellent individual;

Step 3: Using select operator and strategy select the two parents individuals;

Step 4: Using the orthogonal crossover operation

Step 5: Executive the adaptive local search strategy;

Step 6: Executive the adaptive orthogonal mutation strategy for the repeat individuals in the population;

Step 7: Repeat Step 3 to Step 7 until end of the algorithm.

## Experiment and Results

In this section we present the experimental results of the proposed algorithm. All experiments are performed on Intel Core(TM)2 Duo CPU 2.26GHz/4G RAM Laptop. In the experiments all test cases were chosen from TSPLIB (<http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95>). The optimal solution of each test case is known. We list the test cases and their optimal solutions in Table.1.

Table.1 Results of our algorithm

Instance	Result in TSPLIB	Optimum in TSPLIB	Our Result	Run Time
St70	675	678.597	677.129	0.78
Er276	538	545.387	544	1.16
kroA100	21282	21285.443	21285.443	2.69
Rd100	7910	7910.396	7910.396	2.14
Pr136	96772	96772	96771.924	9.11
Pr144	58537	58537	58535.221	7.97

The above results demonstrate clearly the efficiency of the algorithm. Note that for the seven test cases the optimum was found in all ten runs. The number of cities in these test cases varies from 70 to 144. Note also, that the running time of the algorithm was reasonable: below 3 seconds for problems with up to 100 cities, below 10 seconds for the test case of 144 cities.

## Conclusion

In this paper, we introduce a fast genetic algorithm for combinatorial optimization problem. This algorithm is based on a simple genetic algorithm and combine the base ideology of orthogonal test then applied it to the population initialization, crossover operator, as well as the introduction of

adaptive orthogonal local search to prevent local convergence, which have been proved to be very efficient to accelerate the converge speed. The new algorithm shows great efficiency in solving TSP with the problem scale under 300. Particularly, if we choose a comparatively smaller population scale that involves evolution, the algorithm is also efficient to get the approximate best solution in a short executing time.

### Acknowledgement

This paper is supported by the Provincial Natural Science Foundation of Hubei, China (No. 2011CDB334), National Civil Aerospace Pre-research Project of China and Science Research Foundation of Wuhan Institute of Technology (No.12126011).

### References

- [1] J. Holland, Adaptation in natural and artificial systems. University of Michigan press, 1975.
- [2] Durbin R, Willshaw D. An Analogue Approach to the Traveling Salesman Problem Using an Elastic Net Approach. *Nature*, 326, 6114, (1987) 689-691.
- [3] T.Guo and Z.Michalewize. Inver-Over operator for the TSP. In *Parallel Problem Solving from Nature(PPSN V)*, Springer-Verlag press, (1998) 803-812.
- [4] Z.C.Huang, X.L.Hu and S.D.Chen. Dynamic Traveling Salesman Problem based on Evolutionary Computation. In *Congress on Evolutionary Computation(CEC'01)*, IEEE Press, (2001) 1283-1288.
- [5] Aimin Zhou, Lishan Kang and Zhenyu Yan. Solving Dynamic TSP with Evolutionary Approach in Real Time. In *Congress on Evolutionary Computation(CEC'03)*, (2003) 951-957 .
- [6] H.Yang, L.S.Kang and Y.P.Chen. A Gene-pool Based Genetic Algorithm for TSP. *Wuhan University Journal of Natural Sciences* 8(1B), (2003) 217-223.
- [7] X.S.Yan, L.S.Kang et.al: An Approach to Dynamic Traveling Salesman Problem. *Proceedings of the Third International Conference on Machine Learning and Cybernetics*, IEEE Press, Shanghai, (2004) 2418-2420.
- [8] X.S.Yan, A.M Zhou, L.S Kang et.al; TSP Problem Based on Dynamic Environment. *Proceedings of the 5th World Congress on Intelligent Control and Automation*, IEEE press, Hangzhou, China, (2004) 2271-2274.
- [9] Leung Yiu-Wing, Wang Yuping. An orthogonal genetic algorithm with quantization for global numerical optimization. *IEEE Transactions on Evolutionary Computation*, (2001), 5(1): 41-53
- [10] [X.S.Yan, Hui Li et.al: A Fast Evolutionary Algorithm for Combinatorial Optimization Problems. *Proceedings of the Fourth International Conference on Machine Learning and Cybernetics*, IEEE Press, (2005) 3288-3292.
- [11] H.M. Liu et.al. Orthogonal Genetic Algorithm and Its Application in Function Optimization. *Applied Mechanics and Materials Vols. 121-126* (2012) pp 4528-4531.