Available online at www.qu.edu.iq/journalcm
JOURNAL OF AL-QADISIYAH FOR COMPUTER SCIENCE AND MATHEMATICS
ISSN:2521-3504(online) ISSN:2074-0204(print)

# Speeding up Travelling Salesman Problem Using Hybrid Algorithm

***Ali AbdulKadhim Taher*** [a], ***Suhad Malalla Kadhim*** [b]

[a] *Computer Science, University of Technology , Iraq.Email: 0110114@student.uotechnology.edu.iq*

[b] *Computer Science, University of Technology , Iraq.Email: suhad_malalla@yahoo.com*

ARTICLE INFO

*Article history:*
Received: 06 /08/2019
Rrevised form: 21 /08/2019
Accepted : 29 /08/2019
Available online: 25 /09/2019

*Keywords:*

Genetic algorithm,
2-opt algorithm,
Travelling Salesman Problem.

ABSTRACT

Traveling salesman problem (TSP) is one of the most popular optimization problems and attracts great attention from many researchers. Several studies have suggested different approaches to solving two-dimensional TSP. This research will present an enhancement for genetic algorithm by using a 2-opt method to solve TSP, such that we will use a 2-opt method to generate the initial population for genetic algorithm, and the roulette wheel choice strategy, the order crossover operator and 2-opt method for mutation operator will be highlighted .

The experiments of TSP using five real TSP problems taken from the Traveling Salesman Problem Library (TSPLIB), including eil51, st70, pr76,eil76, rd100 (the numbers attached to the problem names represent the number of cities). The error rate of proposed algorithm is (2.233818, 0.030299, 0.057161, 0.084524, 0.123523) . While the error rate of the traditional GA is (2.300088, 0.052921, 0.064016, 0.091775, 0.14666). The results showed the proposed algorithm is acquired higher quality solutions within a reasonable computational time compared to the traditional genetic algorithm. The programming language Pythom3 was used in programming the proposed method .

DOI : 10.29304/jqcm.2019.11.4.622

## 1 . Introduction

Traveling Salesman Problem (TSP) is a well-known problem in the NP-Hard complexity space. The problem has been studied excessively and a wide range of methods have been offered either to find the optimal round or a good approximation takes less time[1].

The 2-opt is a method that emerged in the late 1950s as a combinatorial optimization method to solve the traveling salesman problem[2].

Corresponding author Ali AbdulKadhim Taher

Email addresses: 0110114@student.uotechnology.edu.iq

Communicated by Qusuay Hatim Egaar

Genetic algorithm (GA) has good global search ability through its operation strategies, but it has a slow convergence speed[3].
The goal of this article is to enhance the Genetic algorithm by suggesting a hybrid algorithm between 2-opt and GA to generate populations that can increase efficient solutions. We then apply the proposed approach for solving travelling salesman problem.

## 2. Related work

Kenan Karagul, Kenan Aydemir, and others[4] are combined the 2-Opt algorithms and the harmony search algorithm to solve the traveling salesman problem. In this algorithm real numbers are converted to index values instead of roulette wheel selection operators. The 2-Opt local search algorithm used to improve Solutions. Next, two different parameters from the Traveling Salesman Problem Library (TSPLIB) are tested on the algorithm. The experimental results display that this algorithm offers valuable solutions.
Xin Chen, and Y Zhou[5] was used the complete 2-opt algorithm with Glowworm Swarm Optimization (GSO). In the proposed algorithm, luciferin carriers used in Glowworms were converted to the edge between cities. Modify the probabilistic formula and the luciferin update formula. The complete 2-opt algorithm is implemented to optimize the selected optimal paths every few iterations. The results showed that the proposed algorithm is better than the basic GSO in solving TSP. At the same time, the complete 2-opt algorithm can accelerate the convergence rate.
Partha Sarathi Barma, et al.[6] considered a homogeneous fleet of vehicles. Here a bio-inspired meta-heuristic method named Discrete Ant lion Optimization algorithm (DALO) followed by the 2-opt algorithm for local searching is used to minimize the total routing distance of the Multi-Depot vehicle Routing Problem MDVRP. Here 2-opt algorithm was applied to find the shortest path that begins and ends in the same depot after serving all the cities in the route. The comparison with the Genetic Algorithm, Ant colony optimization, and known best solutions is also discussed and analyzed. This amalgamation of heuristics with local search gives good result in case of MDVRP.

## 3. Travelling Salesman Problem

TSP is considered an optimization problem[7]. It is a problem that is simple to describe but hard to solve since it has been classified as an unresolved problem in polynomial time. This problem belongs to the problem of NP-hard[8]. The main objective of this problem is to find the shortest route to a group of cities starting from a certain city and ending in the same city so that each city is visited only once[9]. This problem is attractive to many researchers because of its simple form. It has shown as an NP-complete problem. An effective way to solve large NP problem has not been found and many problems can be designed by TSP[10].

There are many methods developed to solve the TSP problem, Among these methods are exact algorithms and approximate algorithms. Exact algorithms are implemented to find the perfect solution for all correct solutions in a number of steps. However, it is difficult to apply if a large scale due to exponential complexity. For example, if we have 100 cities, there are about 10155 different solutions[11]. On the other hand, approximate algorithms, especially nature-inspired algorithms, can find acceptable solutions to many NP-hard problems and relatively acceptable run time. These methods are usually simple. For example, these algorithms are the genetic algorithm[2].

## 4. A 2-opt algorithm

There are some round optimization algorithms for the TSP. One of them is the 2-Opt Algorithm, a subclass of the k-Opt algorithm. The 2-Optalgorithm eliminates two edges in a tour, dividing the tour into two paths, then reconnecting these paths in other possible ways and identifying the best ones. The 2-Opt algorithm continues to remove the tour and reconnect it so that improvements cannot be found. The main idea of this method is the random switching between two cities in a particular path to creating a new path with a total distance shorter than the original path[2]. As shown in Figure 1, the locations of cities C and D were changed, the original path changes from A -> B-> C-> D-> A (Fig.1.a) to the new path A-> B-> D -> C-> A (Fig.1.b), thus forming a new path shorter than the original.


Figure1: The path (a)before implementing 2-opt operator (b) after implementing 2-opt operator

The main steps of the 2-opt algorithm are given below:

**Input:** initial tour

**Output:** new shorter tour

**Begin**

**Step1:**Randomize a tour T containing all the n targets , great initial tour.

**Step2:** Evaluate the distance route.

**Repeat :**

Step 3: Choose two edges and switching them.

Step 4 If the length of the new tour is shorter than the initial tour, set the new tour as an initial tour.

**While** ( All the target covered)

**Step5:** Return new tour

**End.**

## 5. Genetic algorithms (GA)

Genetic algorithm (GA) is an optimization and search technique based on the concept of genetics and natural selection. At first, a genetic algorithm selects parents from an initial population of chromosomes to generate offspring using operations such as biological processes, crossover, and mutation. Evaluate all chromosomes using a fitness function to set their fitness. Then, fitness values are used to determine whether the chromosomes are discarded or keep. The less adaptive chromosomes are discarded and the most adaptive ones are kept when generating a new population according to the principle of survival of the fittest. The new population replaces the old. This process is repeated until a specific termination condition satisfied[12]. GA has a good global search capability through its operation strategy but has a slow convergence speed[3]. GA needs longer processing time for a problem with large data are considered as its disadvantages[13].

The main steps of the algorithm are given below:

**INPUT:** Initial population of n chromosomes

**OUTPUT:** Best solution

**Begin**

**Step1:** while termination condition not satisfied do

Evaluate the fitness f(x) of each chromosome x in population

**Repeat**

**Step 2:**Choose at random a pair of parents for mating. Exchange bit strings with crossover to create two offspring.

**Step 3:** Process each offspring by the mutation operator, and insert the resulting offspring in the new population.

**Step 4:** Repeat steps 2 and 3 until all parents are selected and mated (P offspring are created).

**Step 5:** Replace the old population of chromosomes by the new one.

**Step 6:** Evaluate the fitness of each chromosome in the new population.

**Step 7:** Go back to step 3 if the number of generations is less than some upper bound. Otherwise, the final result is the best chromosome created during the search.

End while

**Step8 :** Return best solution

**End**

## 6. The Proposed Algorithm

This paper focuses on speeding up TSP using a hybrid algorithm between GA and a 2-opt method. We will divide the work into two stages, the first stage will focus on generating the initial population by using the 2-opt method and the second stage will improve the solutions by using GA. The initial population is generating by using 2-opt and evaluates their fitness values, the optimal tour made by 2-opt is taken as an initial population for GA, at last, a crossover and mutation will be applied. The termination criterion for the proposed algorithm may be reaching a maximum cycle number. (See algorithm1).

The fitness function in this work is the minimum distance between cities. The total distance of each path for TSP of n cities is calculated according to the coordinates of the cities as follows:

If the coordinates of any two cities i and k are $(x_i, y_i)$ and $(x_k, y_k)$, the distance between them is calculated according eq1.

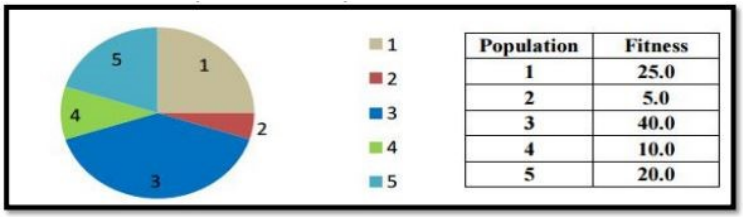
$$D_{ik} = \sqrt{(x_i - x_k)^2 + (y_i - y_k)^2} \quad \text{----------} \quad (1)$$

### 6.1. The 2-opt phase:

In the 2-opt phase, we generate an initial population by exchanging city sites. We will get a set of the population using the process mentioned above. The process will be repeated by the size of the population. We will use this process as well in the mutation operator. (See algorithm 2).

### 6.2. GA phase

Genetic algorithm performs some genetic operators to generate offspring based on the initial population. The genetic process is performed iteratively. The current population is sorted according to its fitness value; roulette wheel operation is used in the proposed algorithm (see figure 2).




| Population | Fitness |
|---|---|
| 1 | 25.0 |
| 2 | 5.0 |
| 3 | 40.0 |
| 4 | 10.0 |
| 5 | 20.0 |

Figure 2: Roulette Wheel selection [14]

Crossover Operator Is an operator that integrates two chromosomes (parents) to create a new chromosome (offspring). The concept of crossover is possible as the new chromosome is better than both parents if it takes the best characteristics of the parents. Ordered crossover used in this algorithm (see figure 3).

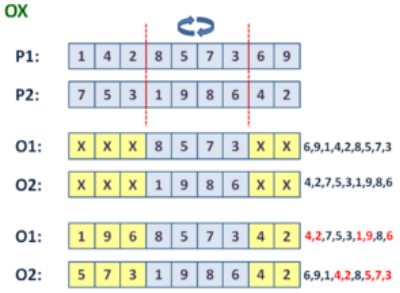



Figure 3: Order crossover[15]

Mutation Operator is to change the genes of the offspring and to increase the diversity of the population. In proposed algorithm we used 2-opt method to generate a new gene. (See algorithm3).

**Algorithm1:** *proposed algorithm*
***Input:*** number of cities , default route
***Output:*** Best solution
**Process:**
**Step1:** Call algorithm2 to generate initial population. /* using 2-opt */
**Step2:** Evaluate the initial population using Eq(1).
**Repeat**
**Step3:** Call algorithm3 that take the initial population and produce new solutions using GA.
**Step 4:** Evaluate the new solutions that were generated from step4 using Eq.(1).
**Step5:** Until the termination condition is met.
**Step6:** Return (best solution).
**End.**

**Algorithm2:** *2-opt phase*
**Input:** initial tour
**Output:** new shorter tour

**Begin**
**Step1:**Randomize a tour T containing all the n targets , great initial tour.
**Step2:** Evaluate the distance route.
**Repeat :**
Step 3: Choose two edges and switching them.
Step 4 If the length of the new tour is shorter than the initial tour, set the new tour as an initial tour.
**While** ( All the target covered)
**Step5:** Return new tour
**End.**

**Algorithm3:** *GA phase*

***Input:*** current population

***Output:*** new solutions

**Process:**

**Step1:** Sort the current population according to their fitness value. /* ascending order */

**Step2:** Select from the current population chromosomes by using roulette wheel according to their fitness.

**Step3:** Perform Crossover Operation between the selected chromosomes generated from step2 to produce new solutions. /* ordered Crossover is used, with crossover rate 0.8 */

**Step 4:** If there is a Mutation rate, perform Mutation operation (Call algorithm2).

**Step5:** Return (new solutions).

## 7.Results and discussion

Experiments were performed to calculate the shortest path, average execution time, and error rate. The rate of the crossover is 0.85 and the rate of mutation is 0.01. All computational experiments were conducted on Intel cores i7-4600U 2.70 GHz machine and coded using python3 programming language.

Some arbitrary data are taken from the Library of TSP (TSPLIB) to be used in the experiment. The data consist of 48, 51, 70, 76 and 100 cities respectively. The Measurements used in these tests is the run time and the length of the shortest path. The simulation is performed 5 times with various recommended parameters.

For the 5 variations of data, we set the number of the population size 48, 51, 70, 76, 100 respectively, and the number of iterations is 5000. The percentage of relative error (%) is calculated using the eq. 2.

$$\text{Error}(\%) = \frac{\text{average solution} - \text{optimal solution}}{\text{optimal solution}} * 100 \quad \text{----------} \quad (2)$$

The results show that this method is relatively superior to the original algorithm. It depends on the best solution, the average solution and the time of each method. The running time of the proposed algorithm is slightly faster than the GA methods. The summary of the simulation is presented in Table1.

The best solution reached in (eil51) where the error rate is (0.03), and the worst solution is satisfied in (att48) where the error rate is (2.23).

Table1: The summary of the simulation.

| Instance | Optimal solution | algorithm | No. of cities | Best tour | Worst tour | Ave. of tour | Error % | Ave. time |
|---|---|---|---|---|---|---|---|---|
| att 48 | 10,628 | GA | 48 | 34278.72 | 35670.27 | 35073.33 | 2.300088 | 80.2716 |
| | | Proposed algorithm | 48 | 34133.55 | 34609.58 | 34369.02 | 2.233818 | 80.071 |
| eil51 | 426 | GA | 51 | 441.8089 | 461.6358 | 448.5443 | 0.052921 | 79.7488 |
| | | Proposed algorithm | 51 | 431.916 | 445.035 | 438.9073 | 0.030299 | 79.441 |
| st70 | 675 | GA | 70 | 697.4344 | 742.4123 | 718.2108 | 0.064016 | 158.934 |
| | | Proposed algorithm | 70 | 690.719 | 732.4991 | 713.5836 | 0.057161 | 154.448 |
| pr76 | 108,159 | GA | 76 | 113664.1 | 119446.8 | 118085.3 | 0.091775 | 185.770 |
| | | Proposed algorithm | 76 | 114861.3 | 119780.4 | 117301 | 0.084524 | 185.530 |
| rd100 | 7910 | GA | 100 | 8922.557 | 9220.022 | 9070.083 | 0.14666 | 287.97 |
| | | Proposed algorithm | 100 | 8647.966 | 9172.798 | 8887.067 | 0.123523 | 280.225 |

## 8. conclusion

A new algorithm has been developed for TSP by combining GA and 2-opt. The key inspiration to develop a hybrid algorithm to utilize from local search in a 2-opt and the global search in the GA, which compensates for the demerits of each algorithm. Thus, a hybrid technique combines the credential of both optimization techniques. As GA and hybrid algorithm computational results are tabulated, hybrid algorithm superior performance on GA for TSP concerning a criterion, i.e., the average mean error and average mean computational time. As well that, the using of the 2-opt for generating the initial population of GA and using 2-opt as mutation method in GA increased the efficiency of the solution.

## References


[1] J. Scholz, "Genetic Algorithms and the Traveling Salesman Problem a historical Review." 2019.

[2] D. S. Johnson and L. A. McGeoch, "The traveling salesman problem: A case study in local optimization," *Local search Comb. Optim.*, vol. 1, no. 1, pp. 215–310, 1997.

[3] C. Liu and L. H. Fan, "Conformal Array Pattern Synthesis Using a Hybrid GA/ABC Algorithm," in *2015 International Conference on Artificial Intelligence and Industrial Engineering*, 2015.

[4] K. Karagul, E. Aydemir, and S. Tokat, "Using 2-Opt based evolution strategy for travelling salesman problem," *An Int. J. Optim. Control Theor. Appl.*, vol. 6, no. 2, pp. 103–113, 2016.

[5] X. Chen, Y. Zhou, Z. Tang, and Q. Luo, "A hybrid algorithm combining glowworm swarm optimization and complete 2-opt algorithm for spherical travelling salesman problems," *Appl. Soft Comput.*, vol. 58, pp. 104–114, 2017.

[6] P. S. Barma, J. Dutta, and A. Mukherjee, "A 2-opt guided discrete antlion optimization algorithm for multi-depot vehicle routing problem," *Decis. Mak. Appl. Manag. Eng.*, 2019.

[7] N. Soni and T. Kumar, "Study of Various Mutation Operators in Genetic Algorithms," *(IJCSIT) International Journal of Computer Science and Information Technology*, vol. 5, no. 3. pp. 4519–4521, 2014.

[8] J.-Y. Potvin, "Genetic algorithms for the traveling salesman problem," *Ann. Oper. Res.*, vol. 63, no. 3, pp. 337–370, 1996.

[9] A. Hassanat and E. Alkafaween, "On enhancing genetic algorithms using new crossovers," *arXiv Prepr. arXiv1801.02335*, 2018.

[10] S. Samanta, A. De, and S. Singha, "SOLUTION OF TRAVELING SALESMAN PROBLEM ON SCX BASED SELECTION WITH PERFORMANCE ANALYSIS USING GENETIC ALGORITHM," *Int. J. Eng. Sci. Technol.*, vol. 3, no. 8, 2011.

[11] K. Helsgaun, "An effective implementation of the Lin–Kernighan traveling salesman heuristic," *Eur. J. Oper. Res.*, vol. 126, no. 1, pp. 106–130, 2000.

[12] S. N. Sivanandam and S. N. Deepa, "Genetic algorithms," in *Introduction to genetic algorithms*, Springer, 2008, pp. 15–37.

[13] Y. Wei, Y. Hu, and K. Gu, "Parallel search strategies for TSPs using a greedy genetic algorithm," in *Third International Conference on Natural Computation (ICNC 2007)*, 2007, vol. 3, pp. 786–790.

[14] A. Sharma and A. Mehta, "Review paper of various selection methods in genetic algorithm," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 3, no. 7, pp. 1476–1479, 2013.

[15] Z. Michalewicz, "Evolution Programs (3ed).PDF." .