BAREKENG: Journal of Mathematics and Its Applications
March 2024 Volume 18 Issue 1 Page 0449–0456
P-ISSN: 1978-7227 E-ISSN: 2615-3017
https://doi.org/10.30598/barekengvol18iss1pp0449-0456

# LEADERS AND FOLLOWERS ALGORITHM FOR TRAVELING SALESMAN PROBLEM

**Helen Yuliana Angmalisang [1*], Syaiful Anam[2]**

[1]*Mathematics Department, Faculty of Mathematics, Natural and Earth Science, Universitas Negeri Manado*
*St. Kampus Unima, Tondano, 95618, Indonesia*
[2]*Mathematics Department, Faculty of Mathematics and Natural Science, Brawijaya University*
*St. Veteran, Malang, 65145, Indonesia*

*Corresponding author's e-mail: * **helenangmalisang@unima.ac.id***

***Article History:***

*Received: 2nd October 2023*
*Revised: 15th December 2023*
*Accepted: 13th January 2024*

***Keywords:***

*Metaheuristics;*
*Leaders and Followers Algorithm;*
*Traveling Salesman Problem;*
*Optimization.*

*ABSTRACT*

*Leaders and Followers algorithm is a metaheuristics algorithm. In solving continuous optimization, this algorithm is proved to be better than other well-known algorithms, such as Genetic Algorithm and Particle Swarm Optimization. This paper aims to apply the Leaders and Followers algorithm for the Traveling Salesman Problem (TSP), a well-known combinatorial optimization problem to minimize distance. There are some modifications in order to fit the algorithm in TSP problems. Some most-used-problems in TSP are used to test this algorithm. The result is that the Leaders and Followers algorithm performs well, stable, and guarantees the optimality of the obtained solution in TSP with fewer than 20 cities. In TSP with a bigger number of cities, the proposed algorithm is not stable and might has difficulties in finding the optimal solutions.*



***How to cite this article:***

H. Y. Angmalisang and S. Anam., "LEADERS AND FOLLOWERS ALGORITHM FOR TRAVELING SALESMAN PROBLEM," *BAREKENG: J. Math. & App.*, vol. 18, iss. 1, pp. 0449-0456, March, 2024.


Journal homepage: *https://ojs3.unpatti.ac.id/index.php/barekeng/*
Journal e-mail: *barekeng.math@yahoo.com; barekeng.journal@mail.unpatti.ac.id*

**Research Article · Open Access**

## 1. INTRODUCTION

Traveling Salesman Problem (TSP) is a problem to find the shortest route to visit all cities exactly once and return to the origin city. It is broadly used in transportation[1] and logistics distributions [2]. However, it is one of the NP-hard problems in combinatorial optimization. As a NP-hard problem, it is unlikely to have an efficient algorithm to compute its exact optimal solution [3]. Thus, it is important to find a method or algorithm which can solve this NP-hard problem quickly and easily.

There have been many approaches in solving TSP, such as Branch and Bound [4], Genetic Algorithm (GA) [5], [6], Tabu Search [6], Particle Swarm Optimization (PSO) [7], Simulated Annealing [8], [9], Ant Colony Optimization (ACO) [10], Symbiotic Organisms Search [11], Producer-Scrounger Method [12], Discreet Social Group Optimization [13]. Clearly, metaheuristics have been widely used for solving TSP. Generally, it is due to the short computation time and effort required by metaheuristics, unlike deterministic or exact methods which require an initial estimate and high mathematical requirements [14]. However, due to its approaching nature, there is no guarantee that the solution obtained from metaheuristics is optimal. Furthermore, since TSP is an NP-hard problem, with the computation time growing exponentially as the size of the problem increases, it is necessary to further investigate whether there are more efficient and effective methods or algorithms.

Leaders and Followers (LaF) algorithm is a novel metaheuristic algorithm proposed by Yasser Gonzalez-Fernandez and Stephen Chen. In solving continuous, unconstrained optimization problems, LaF algorithm is proved to be better than PSO and Differential Evolution (DE) [15]. The LaF algorithm is also better in solving continuous, constrained optimization problem whether it has low or high dimension than Harmony Search (HS), Firefly Algorithm (FA), Cohort Intelligence (CI), Differential Search (DS) and Musical Composition Method (MCM) [16]. Furthermore, when implemented to solve balanced transportation problem [17], LaF algorithm solve faster and more accurate than Vogel's Approximation Method (VAM), Revised Distribution Method (RDI) [18] and The Maximum Range Column Method (MRCM) [19].

This study aims to investigate the performance of the LaF algorithm in solving TSP. Because in the previous studies, the LaF algorithm only addressed continuous optimization [15], [16] and transportation problems [17], which are different from the nature of TSP that is a combinatorial optimization, modifications to the LaF algorithm are needed to effectively solve the TSP. Subsequently, the solutions obtained from the LaF algorithm are compared with those obtained from other metaheuristics.

## 2. RESEARCH METHODS

### 2.1 Standard Leaders and Followers Algorithm

Basically, Leaders and Followers algorithm is based on two principles for effective searching solutions in multi-modal search space: (1) there is no direct comparison between random exploratory sample solutions and the best solutions, because it will lead to local optima and cause premature convergence; and (2) there should be the comparison of two ‘basins’ or set of solutions that has local optima at the earliest stage of searching unless the accumulated information can be isolated.

There are two separated populations, i.e., “Leaders” (*L*) and “Followers” (*F*). These populations serve different roles. *L* serves as the storage of prospective solutions which may be the global optimum, whereas *F* serves in searching for new solutions. The following is the pseudocode of the standard Leaders and Followers algorithm.

Initialize *L* with *n* uniform random vectors.
Initialize *F* with *n* uniform random vectors.
**repeat**
  **for** *i*:= 1 to *n* **do**
    *leader* := Pick an element from *L*.
    *follower* := Pick an element from *F*.
    *trial* := create_trial(*leader*, *follower*)
    **if** *f*(*trial*) <*f*(*follower*) then

Substitute *follower* by *trial* in *F*.
**end if**
**end for**
**if** median($f(F)$) < median($f(L)$) **then**
$L$:= merge_populations($L$, $F$)
Reinitialize $F$ uniformly.
**end if**
**until** The termination criterion is satisfied.

**Algorithm 1.** Pseudocode of Leaders and Followers Algorithm

After initializing the populations uniformly, there are $n$ pairs of *leader* and *follower* which are chosen randomly to create $n$ new solutions, named *trial*. If the *trial* is a better solution than *follower*, then the *follower* is substituted by *trial* in $F$. After that, the median of $L$ and $F$ is compared. If the solution median of $F$ is better than $L$, then select first the best solution in $L$, and after that the algorithm performs binary tournament selection without replacement for the other $n$ - 1 solutions.

### 2.2 Leaders and Followers Algorithm for Traveling Salesman Problem

There should be some modification in order to fit the Leaders and Followers (LaF) algorithm for Traveling Salesman Problem (TSP).

#### 2.2.1 Individuals

Firstly, every individual in the population should represent the route, starting from the initial city, then the next city that will be visited, the one after that represents the next city, this continues up until the last city that will be written in the right end, then go back to the initial city from the last one which is written in the right end. For example, if one of the leaders in L is (1 4 2 3), then it means that the route starts from City 1. From City 1, it goes to City 4, then goes to City 2, and then to City 3. From City 3, it goes back to City 1, the initial city. Thus, every individual must have the same length $n$, because it represents the dimension or the number of cities that should be visited, and consists of integers or characters that represent every city.

#### 2.2.2 Trial

The trial in the standard LaF is generated by the following formula. For every dimension $i$ = 1, 2, …$n$ of the new solution *trial*:

$$trial_i = follower_i + \varepsilon_i 2(leader_i - follower_i) \tag{1}$$

where $\varepsilon_i$ is a uniform random number in (0, 1) sampled independently for every dimension. This kind of generation process is not suitable for combinatorial optimization, especially TSP. Therefore, the *trial* generation in this paper uses a swapping method in a chosen *leader*. Two indexes are randomly chosen, and the elements of route in those indexes are swapped. For example, the chosen *leader* is (1 8 3 5 2 7 6 4), 2 and 5 are randomly chosen indexes. Then, the second city to be visited by the *leader* is swapped with the fifth city. So, the generated *trial* is (1 2 3 5 8 7 6 4).

#### 2.2.3 Cost Function

The cost function in this kind of problem counts the sum of distance of the route from the initial city to the next city, then the next one after it, until going back to the initial city. Thus, the pseudocode of the proposed algorithm is written in **Algorithm 2**.

## 3. RESULTS AND DISCUSSION

The proposed algorithm is evaluated using 6 data sets that are widely used for algorithm evaluation in solving TSP. The data set is acquired from [20]. The data sets have different numbers of cities. *p01* data set has 15 cities, *burma14* data set has 14 cities, *gr17* data set has 17 cities, *fri26* data set has 26 cities, *dantzig42* data set has 42 cities, and *att48* data set has 48 cities.

For each data set, the algorithm is run for 20 times to obtain the best, worst, and average solution. The algorithm is coded in MATLAB. Upon a series of experiment conducted by the authors, a set of effective parameter values is found to be as 200 for population size if $n < 20$ and 3000 if $n > 20$. The comparisons of results by proposed algorithm and other algorithms.

```
Initialize L with n uniform random integer vectors without repetition.
Initialize F with n uniform random integer vectors without repetition.
repeat
    for i:= 1 to n do
        leader := Pick an element from L.
        follower := Pick an element from F.
        a:=randomly pick an integer in [1, n]
        b:= randomly pick an integer in [1, n]
        trial:= leader
        swap(trial(a), trial(b))
        if f(trial) <f(follower) then
            Substitute follower by trial in F.
        end if
    end for
    if median(f(F)) < median(f(L)) then
        L := merge_populations(L, F)
        Reinitialize F uniformly.
    end if
until the termination criterion is satisfied.
```

Algorithm 2. Pseudocode of Leaders and Followers Algorithm for TSP

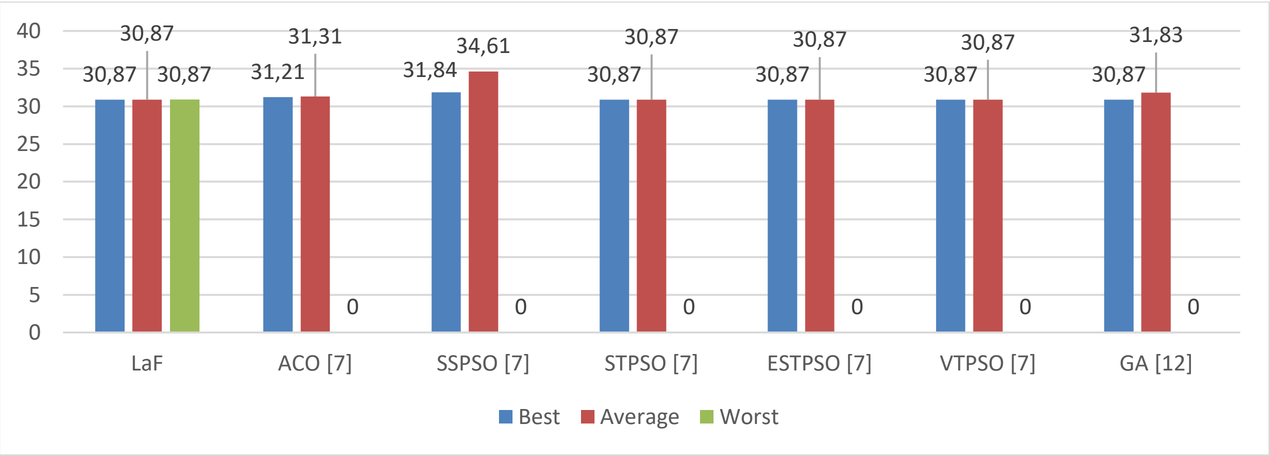


**Figure 1. The Comparison of Solutions by Proposed Algorithm (LaF) and Other Algorithms for burma14**

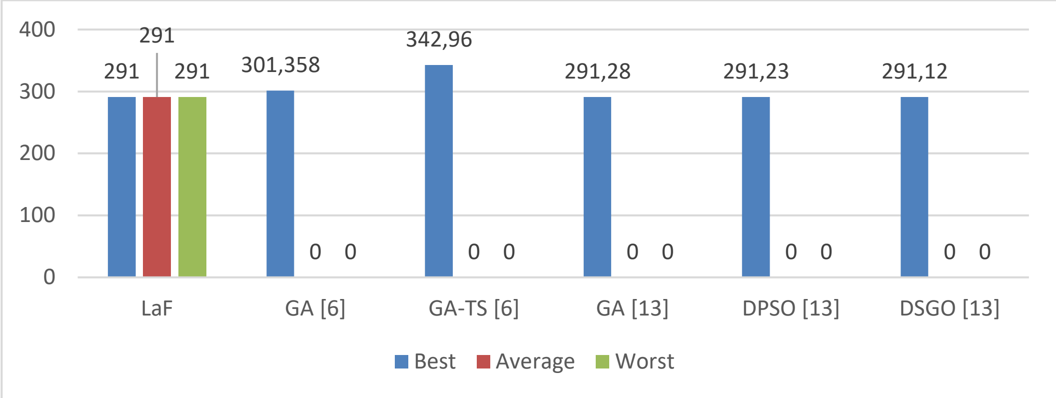


**Figure 2. The Comparison of Solutions by Proposed Algorithm (LaF) and Other Algorithms for p01**

In the first TSP case, *burma14*, the solution obtained by the proposed algorithm (LaF) is compared to Ant Colony Optimization (ACO) [7], [12], Swap Sequence based PSO (SSPSO) [7], Self-Tentative PSO (STPSO) [7], Enhanced Self-Tentative PSO (ESTPSO) [7], Velocity Tentative PSO (VTPSO) [7], Genetic Algorithm [12], and Producer-Scrounger Method (PSM) [12]. The optimal solution in this case is 30.87. The result comparison for this case is shown on Figure 1. From the comparison, it is shown that LaF is stable in finding the optimal solution. Since it is relatively easy, because the dimension ($n$) is low, LaF, STPSO, ESTPSO, and VTPSO can find the optimal solution in each test. However, ACO and SSPSO cannot find the optimal solution, whereas GA and PSM are not stable, since they cannot find the optimal solution in some tests.

In the second case, *p01*, the solution obtained by the proposed algorithm (LaF) is compared to Genetic Algorithm (GA) [6], Genetic Algorithm-Tabu Search (GA-TS) [6], Discrete Particle Swarm Optimization (DPSO) [13], and Discreet Social Group Optimization (DSGO) [13]. The optimal solution in this case is 291. The result comparison for this case is shown on Figure 2. From the comparison, it is shown that LaF is stable in finding the optimal solution and better than the other algorithm, because they cannot find the optimal solution.

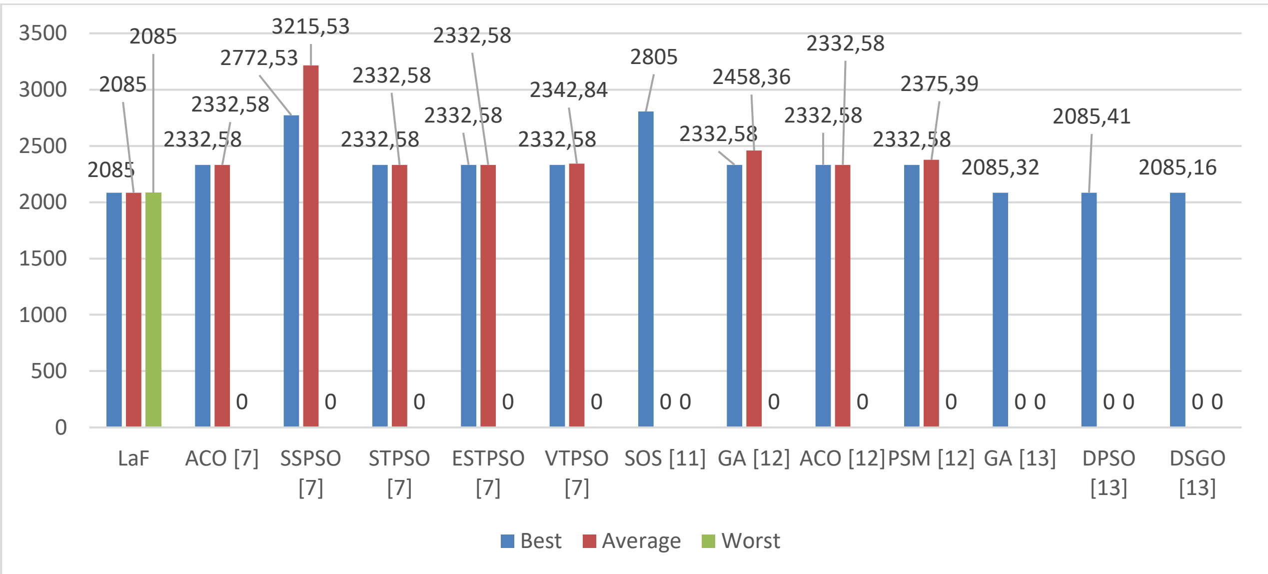


**Figure 3. The Comparison of Solutions by Proposed Algorithm (LaF) and Other Algorithms for gr17**

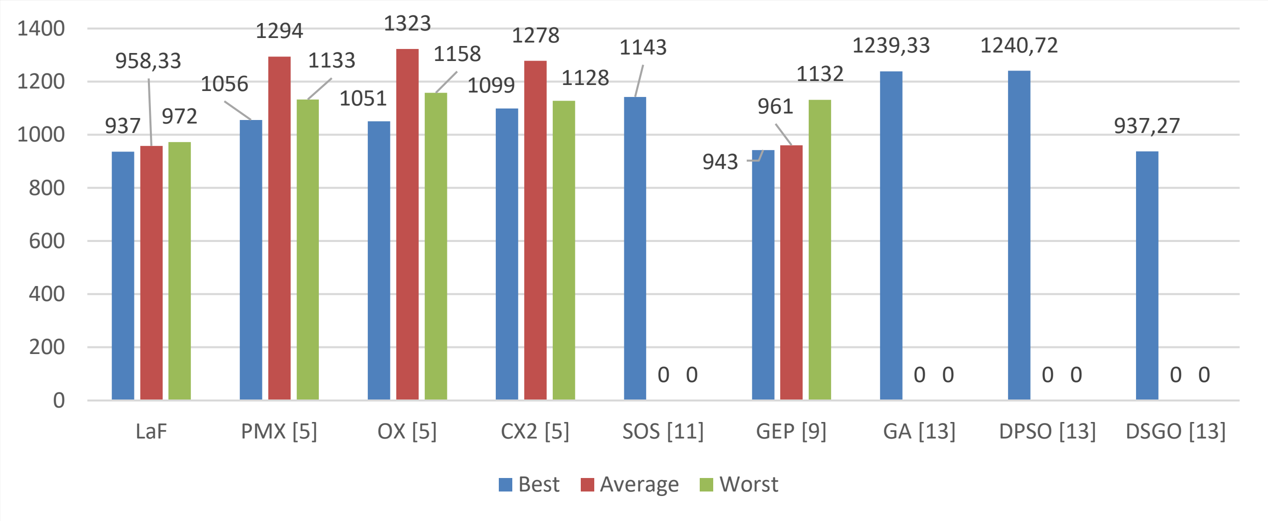


**Figure 4. The Comparison of Solutions by Proposed Algorithm (LaF) and Other Algorithms for fri26**

In the third case, *gr17*, the solution obtained by the proposed algorithm (LaF) is compared to ACO [6], SSPSO [7], STPSO [7], ESTPSO [7], VTPSO [7], Symbiotic Organisms Search (SOS) [11], GA [12], [13], ACO [12], PSM [12], GA [13], DPSO [13], and DSGO [13]. The optimal solution in this case is 2085. The result comparison for this case is shown on Figure 3. From the comparison, it is shown that LaF is also stable in finding the optimal route. LaF always finds the optimal route in every test for this problem. SOS [11] also finds the optimal solution. However, there is no information about the average and worst solutions obtained by SOS. Except LaF and SOS, Figure 3 shows that all algorithms cannot find the optimal solution. Thus, LaF is better in this case.

In the fourth case, *fri26*, the solution obtained by the proposed algorithm (LaF) is compared to Genetic Algorithm with Partially Mapped Crossover (PMX) [5], Genetic Algorithm with Order Crossover (OX) [5], Genetic Algorithm with New Cycle Crossover Operator (CX2) [5], SOS [11], Gene-Expression Programming (GEP) [9], GA [13], DPSO [13], and DSGO [13]. The optimal solution in this case is 937. The result comparison for this case is shown on Figure 4. It is shown that LaF finds optimal solutions, but is not stable. It can only find the optimal solution in a few trials. However, LaF is still much better than the other algorithms, except DSGO. The best solution of DSGO approaches optimal, but since there is no information about its average and worst solution, it cannot be concluded whether it is better than LaF or not in this case.

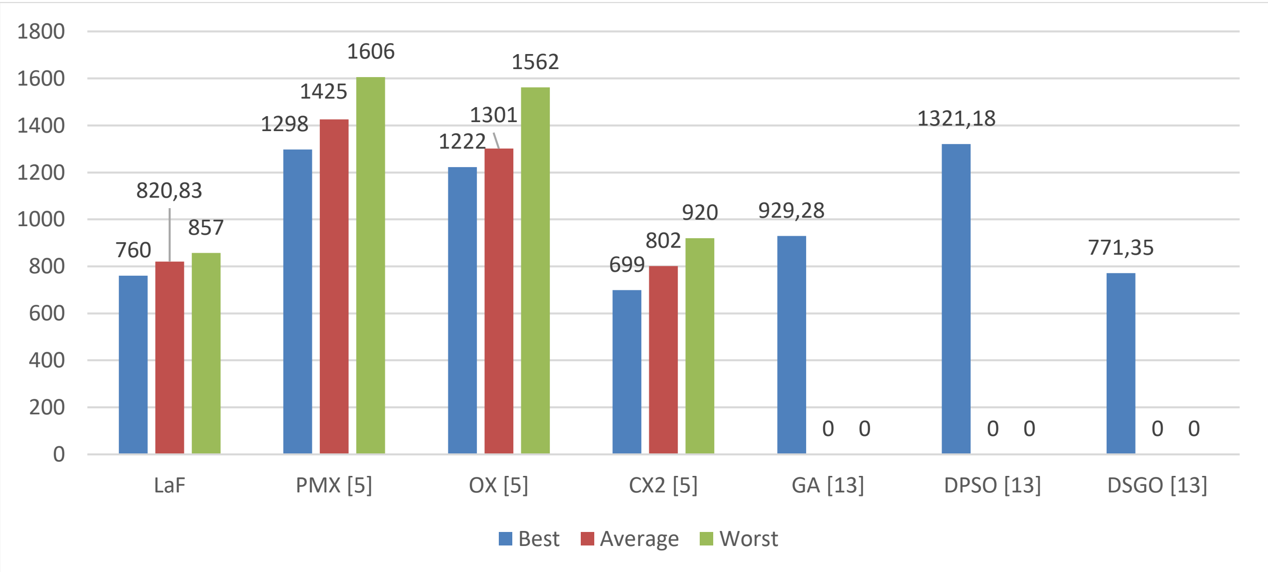


**Figure 5. The Comparison of Solutions by Proposed Algorithm (LaF) and Other Algorithms for dantzig42**

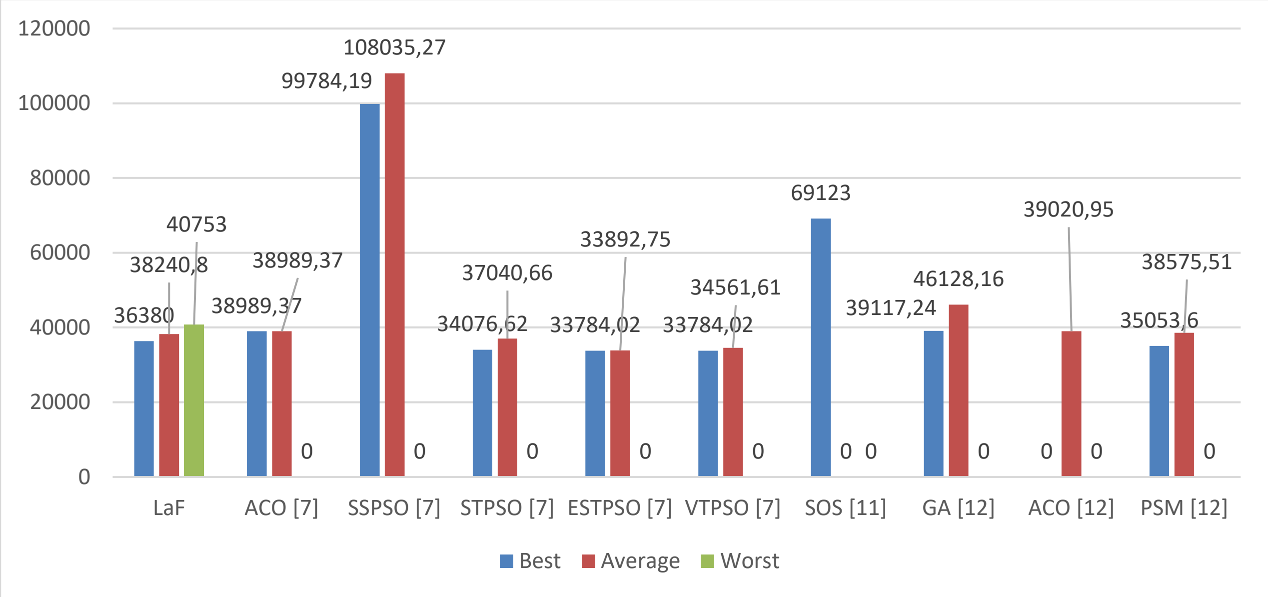


**Figure 6. The Comparison of Solutions by Proposed Algorithm (LaF) and Other Algorithms for att48**

In the fifth case, *dantzig42*, the solution obtained by the proposed algorithm (LaF) is compared to PMX[5], OX [5], CX2 [5], GA [13], DPSO [13], and DSGO [13]. The optimal solution in this case is 699. The result comparison for this case is shown on Figure 5. It is shown that LaF obtains better solutions than other algorithms, except CX2 which obtains better solutions in its best and average.

In the last case, *att48*, the solution obtained by the proposed algorithm (LaF) is compared to ACO [6], SSPSO [6], STPSO [6], ESTPSO [6], VTPSO [6], SOS [10], GA [11], ACO [11], and PSM [11]. The optimal solution in this case is 33523. The result comparison for this case is shown on Figure 6. From the result, it is shown that STPSO [6], ESTPSO [6], VTPSO [6] are better than LaF. However, LaF is still better than the rest of the algorithms.

Based on the evaluation and comparison result, it is shown that the Leaders and Followers algorithm performs well and stable when applied to the Traveling Salesman Problem with fewer than 20 cities. Overall, it is better than other algorithms, namely GA, GA-TS, ACO, SSPSO, STPSO, ESTPSO, SOS, VTPSO, and PSM, DPSO, and DSGO. Even though DSGO approaches optimal in *fri26*, but in other cases, LaF consistently obtains the optimal solution which is better than DSGO, like in *p01* and *gr17*.

For the problem with a huge number of cities, i.e. > 40 cities, the results show that LaF has difficulties in reaching the optimal solutions. However, it still obtains much better solutions than other algorithms, except CX2, STPSO, ESTPSO, and VTPSO. Nevertheless, generally, it cannot be concluded that CX2, STPSO, ESTPSO, and VTPSO are better than LaF, because they have much worse solutions in *burma14*, *gr17, and fri26*.

## 4. CONCLUSIONS

In this study, the Leaders and Followers algorithm are applied for solving the Traveling Salesman Problem. There are some modifications needed to fit the algorithm, such as its individual formation, trial creation, and cost function. After being evaluated using some widely used data sets, it is concluded that the Leaders and Followers algorithm performs well, stable, fast, and guarantees the optimality of the obtained solution in TSP with fewer than 20 cities. It is unlike some of the other algorithms which are not stable and converge prematurely, so they cannot find optimal solutions in many experiments. However, in TSP with more than 20 cities, the proposed algorithm is not stable and might has difficulties in the finding optimal solutions.

## REFERENCES


[1] K. Stoilova and T. Stoilov, "Transportation modelling and solving Travelling Salesman problem," *IOP Conf Ser Mater Sci Eng*, vol. 878, no. 1, p. 12026, Jun. 2020, doi: 10.1088/1757-899X/878/1/012026.

[2] M. Azizur Rahman, A. Islam, and L. Ershad Ali, "Intelligent Route Construction Algorithm for Solving Traveling Salesman Problem," *IJCSNS International Journal of Computer Science and Network Security*, vol. 21, no. 4, p. 33, 2021, doi: 10.22937/IJCSNS.2021.21.4.5.

[3] D.-Z. Du, P. Pardalos, X. Hu, and W. Wu, *Introduction to Combinatorial Optimization*. 2022. doi: 10.1007/978-3-031-10596-8.

[4] G. Finke, A. Claus, and E. Gunn, "A two-commodity network flow approach to the traveling salesman problem," *Congress Num*, vol. 41, Oct. 1984.

[5] A. Hussain, Y. S. Muhammad, M. Nauman Sajid, I. Hussain, A. Mohamd Shoukry, and S. Gani, "Genetic Algorithm for Traveling Salesman Problem with Modified Cycle Crossover Operator," *Comput Intell Neurosci*, vol. 2017, 2017, doi: 10.1155/2017/7430125.

[6] S. Elza Ramadhania and S. Rani, "Implementasi Kombinasi Algoritma Genetika dan Tabu Search untuk Penyelesaian Travelling Salesman Problem."[Online]. Available: http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsp/

[7] A. MAH, S. I. Hossain, and S. Akter, "A Comparative Study of Prominent Particle Swarm Optimization Based Methods to Solve Traveling Salesman Problem," *International Journal of Swarm Intelligence and Evolutionary Computation*, vol. 05, no. 03, 2016, doi: 10.4172/2090-4908.1000139.

[8] A. R. BOTSALI and K. ALAYKIRAN, "Analysis of TSP: Simulated Annealing and Genetic Algorithm Approaches," *International Journal of Computational and Experimental Science and Engineering*, vol. 6, no. 1, pp. 23–28, Mar. 2020, doi: 10.22399/ijcesen.637445.

[9] A. H. Zhou *et al.*, "Traveling-salesman-problem algorithm based on simulated annealing and gene-expression programming," *Information (Switzerland)*, vol. 10, no. 1, Dec. 2018, doi: 10.3390/info10010007.

[10] K. Thirugnanasambandam, R. S. Raghav, D. Saravanan, U. Prabu, and M. Rajeswari, "Experimental analysis of ant system on travelling salesman problem dataset TSPLIB," *EAI Endorsed Trans Pervasive Health Technol*, vol. 5, no. 19, 2019, doi: 10.4108/eai.13-7-2018.163092.

[11] M. Isnaini, H. Umam, B. Santosa, and N. Siswanto, *Prosiding Seminar Nasional Manajemen Teknologi XXIV Program Studi MMT-ITS, Surabaya 23 Januari 2016 MODIFIKASI ALGORITMA SYMBIOTIC ORGANISMS SEARCH UNTUK TRAVELING SALESMAN PROBLEM*.

[12] M. A. H. Akhand, P. C. Shill, Md. F. Hossain, A. B. M. Junaed, and K. Murase, "Producer-Scrounger Method to Solve Traveling Salesman Problem," *International Journal of Intelligent Systems and Applications*, vol. 7, no. 3, pp. 29–36, Feb. 2015, doi: 10.5815/ijisa.2015.03.04.

[13] S. Verma, J. J. Jena, C. Satapathy, and M. Rout, "Solving Travelling Salesman Problem using Discreet Social Group Optimization," 2020. [Online]. Available: https://people.sc.fsu.edu/~jburkardt/datasets/tsp/tsp.ht

[14] H. Garg, "A hybrid PSO-GA algorithm for constrained optimization problems," *Appl Math Comput*, vol. 274, pp. 292–305, Feb. 2016, doi: 10.1016/j.amc.2015.11.001.

[15] Y. Gonzalez-Fernandez and S. Chen, "Leaders and followers-A new metaheuristic to avoid the bias of accumulated information," in *2015 IEEE Congress on Evolutionary Computation, CEC 2015 - Proceedings*, Institute of Electrical and Electronics Engineers Inc., Sep. 2015, pp. 776–783. doi: 10.1109/CEC.2015.7256970.

[16] H. Y. Angmalisang, S. Anam, and S. Abusini, "Leaders and followers algorithm for constrained non-linear optimization," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 13, no. 1, pp. 162–169, Jan. 2019, doi: 10.11591/ijeecs.v13.i1.pp162-169.

[17] H. Y. Angmalisang, H. Angmalisang, and S. J. A. Sumarauw, "Leaders and Followers Algorithm for Balanced Transportation Problem," *Computer Engineering and Applications*, vol. 12, no. 2, 2023, Accessed: Oct. 01, 2023. [Online]. Available: https://comengapp.unsri.ac.id/index.php/comengapp/article/view/436/265

[18] S. Aramuthakannan, "Revised Distribution Method of finding Optimal Solution for Transportation Problems," *IOSR Journal of Mathematics*, vol. 4, pp. 39–42, Oct. 2013, doi: 10.9790/5728-0453942.

[19] H. B. Kalhoro, H. Abdulrehman, M. M. Shaikh, and A. S. Soomro, "THE MAXIMUM RANGE COLUMN METHOD-GOING BEYOND THE TRADITIONAL INITIAL BASIC FEASIBLE SOLUTION METHODS FOR THE TRANSPORTATION PROBLEMS," 2021.[Online]. Available: https://doi.org/00.00000/jmcms.2021.01.00006

[20] "Data for the Traveling Salesperson Problem." Accessed: Oct. 01, 2023.[Online]. Available: https://people.sc.fsu.edu/~jburkardt/datasets/tsp/tsp.html