Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Issue 1, January 2023
DOI : https://doi.org/10.33395/sinkron.v8i1.12016
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Combination of Ant Colony Tabu Search Algorithm with Firefly Tabu Search Algorithm (ACTS-FATS) in Solving the Traveling Salesman Problem (TSP)

Siti Sarah Harahap[1)*], Poltak Sihombing [2)], Muhammad Zarlis [3)]
[1,2)]Master of Informatics Program, Faculty of Computer Science and Information Technology, Universitas Sumatera Utara, Medan, Indonesia,
[3]Department of Information Systems, Bina Nusantara University
[1)] Sarahharahap0409@gmail.com, [2)] Poltak@usu.ac.id, [3)]Muhammad.zarlis@binus.edu

Submitted : Dec 16, 2022 | **Accepted** : Dec 18, 2022 | **Published** : Jan 1, 2023

Abstract: Traveling Salesman Problem (TSP) is a classic combinatorial optimization problem, one of the optimization problems that can be applied to various activities such as finding the shortest path. The optimization problem in TSP is the most widely discussed and has become the standard for testing computational algorithms. TSP is a good object to test optimization performance. With scientific developments in the field of informatics, many researchers have optimized the application of algorithms to solve the Traveling Salesman Problem (TSP). In this study, researchers used a combination of Ant Colony Tabu Search – Firefly Algorithm Tabu Search (ACTS-FATS). The combination is doneto overcome Premature Convergence (trapped local optimum) which is a shortcoming of the ant colony algorithm, get the best running time by looking at the process of each point movement with the ant colony and firefly methods. After testing, getting the best running time results of 27.79%, and getting an accuracy rate of 17%.

Keywords: ACTS-FATS, Traveling Salesman Problem

INTRODUCTION

In the era of globalization, technology plays a very important role in human life to make it easier to carry out various activities. Progress in education, health, transportation, communication, and other fields is an example that humans need technology in their daily lives. In research case studies, many researchers are interested in taking research topics regarding optimization and hybrid systems (Dewantoro et al., 2019). This is because these topics can be developed, combined, or added to improve the performance results of each processing (Hartono, Mhd. Furqan, Tulus, 2015). Optimization is one of the problems that is often used in testing the performance of a method. The roles obtained from optimization studies can be implemented in terms of planning, scheduling and searching for scientific fields (Septiyafi et al., 2019). Whereas in a hybrid system, researchers make improvements, mergers or an updated system to see how the performance from all sides of the implementation of the hybrid system is.

In connection with this, a thought was proposed to analyze the combination of 2 methods. This combination can overcome premature convergence (stuck in local optimum) (Dewantoro et al., 2019), and is able to provide alternative route solutions that can be used if there are obstacles on the main route which hinders the journey to the destination as well as achieving better running time and obtaining increased accuracy from previous research.

LITERATURE REVIEW

Traveling Salesman Problem (TSP)

The Traveling Salesman Problem (TSP) is an optimization problem that can be applied to various activities such as finding the shortest path. The optimization problem in TSP is the most widely discussed and has become the standard for testing computational algorithms (Min et al., 2019). The main problem with TSP is that a salesman has to visit all cities with known distances between cities and return to his hometown (Deng et al., 2019).

To assess whether we are making progress in solving TSP problems, we must judge from the decreasing amount of time. For example we have a new method A which is faster to solve TSP problems than method B, we

*name of corresponding author

e-ISSN : 2541-2019
p-ISSN : 2541-044X

will judge that we have found a better solution (Alaidi & Mahmood, 2018). But the ranking problem for this method will be very difficult, because methods that are very closely related to each other cannot be judged only through simple comparisons (mild TSP problems) (X. Yang & Wang, 2016). The problem faced by TSP is how to plan the minimum total distance. To solve this problem is not easy because there is a search space from a set of permutations of a number of cities. the distance between the two cities can be calculated by the following equation where dA,B= distance from city A and city B andx, y = Point coordinates (City).

$$d_{A,} = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2} \qquad (1)$$

Ant Colony Optimization (ACO)

The ACO algorithm was introduced by Marco Dorigo (Meng et al., 2019). ACO is a metaheuristic method inspired by the intelligence of ants in finding the shortest path to a food source. ACO applies an optimization problem solving method based on the principle of communication between ant colonies. Basically, all ants will leave a trail of special substances known as pheromones (Alobaedy et al., 2017). This pheromone will then become a guideline for other ants in searching.

Firefly Algorithm (FA)

The firefly algorithm (FA) is a metaheuristic algorithm inspired by the flashing behavior of fireflies (X. S. Yang, 2010). There are two basic functions of the flashing light, namely to attract the attention of other fireflies (communication) and to attract prey (Hay's, 2017a). The Tabu Search algorithm is one of the algorithms that are within the scope of the heuristic method (Dewantoro et al., 2019). This algorithm uses short-term memory to keep the search process from getting stuck at the local optimum value. The Tabu Search algorithm uses a tabu list to store a set of recently evaluated solutions (Gadioli et al., 2018).

Tabu Search

The Tabu Search algorithm is one of the algorithms that are within the scope of the heuristic method (Dewantoro et al., 2019). This algorithm uses short-term memory to keep the search process from getting stuck at the local optimum value (Hay's, 2017b). The Tabu Search algorithm uses a tabu list to store a set of recently evaluated solutions (Gadioli et al., 2018). During the optimization process in each iteration, the solution to be evaluated will be matched first with the contents of the tabu list to see if the solution is already in the tabu list.

METHOD

The stages of the research to be carried out are shown in Figure 1

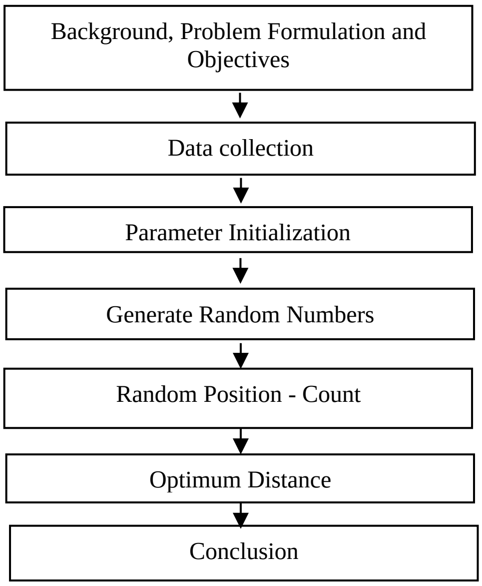

Figure 1. Research Stages

In solving the Traveling Salesman Problem, there are many methods that support searching for a place. One of these methods will be discussed in this study, namely a combination of the two methods that have been studied by previous researchers. Discussion to be carried outto maximize the performance of the combination of the two Ant Colony - Tabu Search (ACTS) and Firefly Algorithm - Tabu Search (FATS) methods, it is hoped that these results will be able to provide optimal distances and produce the best running time in solving the Traveling Salesman Problem (TSP).

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Issue 1, January 2023
DOI : https://doi.org/10.33395/sinkron.v8i1.12016
e-ISSN : 2541-2019
p-ISSN : 2541-044X

The dataset used is Hasler Whitney 3 TSPLIB95, Eil51 and Eil76 data. The dataset can be accessed publicly on the TSPLIB website. Furthermore, the data is processed using the ACTS and FATS algorithms and conclusions will be drawn on the optimal distance andthe value of the running time in the algorithm. The following are the research steps.

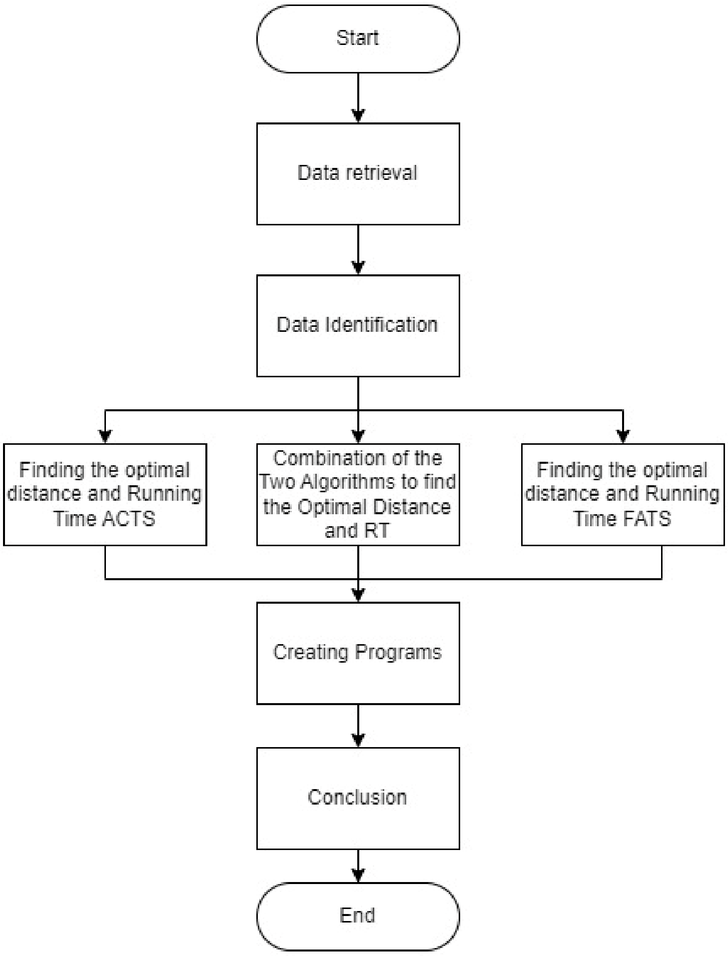

Figure 2. Research Steps

RESULT

Calculation of the ACTS algorithm in finding the shortest route in the TSP problem from the routes in the dataset as follows. Calculating the distance between cities according to the dataset coordinates with the equation:

$$d_{rs} = \sqrt{(x_r - x_s)^2 + (y_r - y_s)^2} \quad (2)$$

To calculate the visibility between nodes is the inverse of the distance (d) as a medium of information quality of a node with the equation:

$$Visibilitas\ (\eta_{rs}) = \frac{1}{d_{rs}} \quad (3)$$

Table 1. City Variable ACTS - FATS

City	X	Y
1	212	169
2	204	169
3	196	169
4	188	169
5	196	161
6	188	154

*name of corresponding author

After determining the distance between cities in the dataset, the possible routes are determined using the tabu search algorithm as follows:

Iteration 1 :

Alternate route search for mileage and visibility

Initial path:

1-5-2-3-6-4-1 with mileage is 110.4 and visibility is 0.42.

Taboo list : 1-5-2-3-6-4-1.

Table 2. Travel Tablist

No	Travel route	Mileage	Visibility
1	1-2-5-3-6-4-1	100.6	0.489
2	1-3-2-5-6-4-1	101.19	0.4425
3	1-6-2-3-5-4-1	145.53	1,258
4	1-4-2-3-6-5-1	108.97	0.408
5	1-5-3-2-6-4-1	104.86	0.461
6	1-5-6-3-2-4-1	108.97	0.408
7	**1-5-4-3-6-2-1**	**96.46**	**1,233**
8	1-5-2-6-3-4-1	112.46	0.387
9	1-5-2-4-6-3-1	110.4	0.348
10	1-5-2-3-4-6-1	143.27	0.414

Table 3. ACTS and FATS Inter-City Distance Matrix

City	1	2	3	4	5	6
1	0	8	16	24	17.8	71.16
2	8	0	8	16	11.31	23.06
3	16	8	0	11	8	25.29
4	24	16	11	0	11.31	24
5	17.8	11.31	8	11.31	0	17.88
6	71.16	23.06	25.29	24	17.88	0

Table 4. ACTS and FATS Inter-City Visibility Matrix

City	1	2	3	4	5	6
1	0	0.125	0.0625	0.071	0.056	0.014
2	0.125	0	0.125	0.062	0.088	0.043
3	0.062	0.125	0	0.09	0.125	0.039
4	0.071	0.062	0.09	0	0.88	0.041
5	0.056	0.088	0.125	0.88	0	0.055
6	0.014	0.043	0.039	0.041	0.055	0

All d pheronom values at the beginning of the calculation are assigned very small numbers. In this calculation, the initial pheromone value uses valueτinitial of 0.01. Determining the initial value is intended so that each route has an interest value to be visited by each ant. Can be seen in the matrix on the following page:

Table 5. Every City Pheronome

*name of corresponding author

DOI : https://doi.org/10.33395/sinkron.v8i1.12016
e-ISSN : 2541-2019
p-ISSN : 2541-044X

City/τ	1	2	3	4	5	6
1	0	0.01	0.01	0.01	0.01	0.01
2	0.01	0	0.01	0.01	0.01	0.01
3	0.01	0.01	0	0.01	0.01	0.01
4	0.01	0.01	0.01	0	0.01	0.01
5	0.01	0.01	0.01	0.01	0	0.01
6	0.01	0.01	0.01	0.01	0.01	0

Find the highest probability value with the formula:

$$probabilitas = \frac{[T_{ij}(\text{t})]^{\alpha}[\eta_{ij}]^{\beta}}{\Sigma[T_{ij}(\text{t})]^{\alpha}[\eta_{ij}]^{\beta}} \quad (4)$$

$$\text{Count}[Tij(t)]^{\alpha}[nij]^{\beta} \quad (5)$$

The pheromone change is called the local pheromone change which can be calculated by the formula:

$$\tau_{(r,s)} = (1-\rho).\tau_{(r,s)} + \rho.\Delta\tau_{(r,s)} \quad (6)$$

$$\Delta\tau_{(r,s)} = \text{change } pheromone = 0 \text{ siklus } = 1 (7)$$

Displays the search results from the first cycle as shown in Table 6 below:

Table 6. Cycle 1 Best Route

Ant	Travel route	Route Distance	Total Mileage
1	1 – 6 – 3 – 4 – 2 – 5 – 1	71.16 - 25.29 - 11 - 16 - 11.31 - 17.8	119.27
2	1 – 6 – 5 – 4 – 2 – 3 – 1	71.16 - 17.88 – 11.31 - 16 - 8 - 16	140.35
3	1 – 6 – 5 – 4 – 3 – 2 – 1	71.16 – 17.88 – 11.31 – 11 – 8 – 8	119.35
4	1 – 6 – 4 – 5 – 3 – 2 – 1	71.16 – 24 – 11.31 – 8 – 8 – 8	130.47
5	1 – 6 – 3 – 5 – 4 – 2 – 1	71.16 – 25.29 – 8 – 11.31 – 16 – 8	139.76
6	**1 – 3 – 6 – 5 – 4 – 2 – 1**	**16 – 26.29 – 17.88 – 11.31 – 16 - 8**	**95.48**

The light intensity level of a firefly x can be seen in the formula:

$$I\ (x) = \frac{1}{f\ (x)} \quad (8)$$

The distance between fireflies every i and j, with the formula:

$$r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (9)$$

Movement of fireflies:

$$x_i = x_i + \beta_0 \times exp^{-\gamma r^2} \times \left(x_j - x_i\right) + \alpha \times \left(rand - \tfrac{1}{2}\right) (10)$$

Iteration 1 :

Table 7. Trial Iteration 1

POPULATION	xi	f(x)	i(i)
x1	119.27	14225.3329	14225.3329
x2	140.35	19698.1225	19698.1225
x3	119.35	14244.4225	14244.4225
x4	130.47	17022.4209	17022.4209
x5	139.76	19532.8576	19532.8576
x6	95.48	8926.4704	8926.4704

Table 8. FATS Movement 1

i=1	i=2	i=3	i=4	i=5	i=6

*name of corresponding author

SinkrOn
JURNAL & PENELITIAN TEKNIK INFORMATIKA
Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Issue 1, January 2023
DOI : https://doi.org/10.33395/sinkron.v8i1.12016
e-ISSN : 2541-2019
p-ISSN : 2541-044X

-	-	-	-	-	Movement
Movement	-	Movement	Movement	Movement	Movement
Movement	-	-	-	-	Movement
Movement	-	Movement	-	-	Movement
Movement	-	Movement	Movement	-	Movement
-	-	-	-	-	-

Table 9. FATS Firefly Movement 1

iteration 1	**I=1**	**r**	**Xnew**	**rand**	**f(x)**	**i(i)**	
x1	permanent	0	119.1526544	0.304424025	14197.35505	14197.35505	
x2	Movement	21.08	119.0273456	0.095575975	14167.509	14167.509	
x3	Movement	0.08	119.457293	0.812154918	14270.04484	14270.04484	
x4	Movement	11.2	119.4390802	0.78180027	14265.69387	14265.69387	
x5	Movement	20.49	119.4869629	0.861604766	14277.13429	14277.13429	best
x6	permanent	24.79	119.1869629	0.361604766	14205.53212	14205.53212	
	i=2	**r**	**Xnew**	**rand**	**f(x)**	**i(i)**	
x1	permanent	21.08	140.2059621	0.259936774	19657.7118	19657.7118	
x2	permanent	0	140.4514367	0.669061084	19726.60606	19726.60606	best
x3	permanent	21	140.0962207	0.077034437	19626.95104	19626.95104	
x4	permanent	9.88	140.4488135	0.664689245	19725.86923	19725.86923	
x5	permanent	0.59	140.2084709	0.264118085	19658.4153	19658.4153	
x6	permanent	45.87	140.4069104	0.594850716	19714.1005	19714.1005	

Table 10. FATS 1 Firefly Movement (Continued)

	i=3	**r**	**Xnew**	**rand**	**f(x)**	**i(i)**	
x1	permanent	0.08	119.2945612	0.407601944	14231.19232	14231.19232	
x2	Movement	21	119.5405002	0.817500391	14289.9312	14289.9312	best
x3	permanent	0	119.1791237	0.215206085	14203.66351	14203.66351	
x4	Movement	11.12	119.4036996	0.589499334	14257.24348	14257.24348	
x5	Movement	20.41	119.2057312	0.25955199	14210.00635	14210.00635	
x6	permanent	24.87	119.4010286	0.585047687	14256.60563	14256.60563	
	i=4	**r**	**Xnew**	**rand**	**f(x)**	**i(i)**	
x1	permanent	11.2	130.4967968	0.544661274	17029.41397	17029.41397	
x2	Movement	9.88	130.3433887	0.288981105	16989.39897	16989.39897	
x3	permanent	11.12	130.5124766	0.570794391	17033.50656	17033.50656	
x4	permanent	0	130.5131677	0.571946161	17033.68694	17033.68694	
x5	Movement	9.29	130.4995765	0.54929423	17030.13948	17030.13948	best
x6	permanent	35.99	130.1975767	0.045961089	16951.40897	16951.40897	
	i=5	**r**	**Xnew**	**rand**	**f(x)**	**i(i)**	
x1	permanent	20.49	139.8693516	0.682252746	19563.43553	19563.43553	

*name of corresponding author

SinkrOn
JURNAL & PENELITIAN TEKNIK INFORMATIKA
Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Issue 1, January 2023
DOI : https://doi.org/10.33395/sinkron.v8i1.12016
e-ISSN : 2541-2019
p-ISSN : 2541-044X

x2	Movement	0.59	139.7974517	0.562419479	19543.3275	19543.3275	best
x3	permanent	20.41	139.8535531	0.655921909	19559.01633	19559.01633	
x4	permanent	9.29	140.0157639	0.926273212	19604.41415	19604.41415	
x5	permanent	0	139.4945248	0.057541351	19458.72245	19458.72245	
x6	permanent	45.28	139.6019213	0.236535526	19488.69644	19488.69644	
	i=6	**r**	**Xnew**	**rand**	**f(x)**	**i(i)**	
x1	Movement	24.79	94.77804309	0.996738481	8982.877452	8982.877452	best
x2	Movement	45.87	94.40554899	0.375914979	8912.40768	8912.40768	
x3	Movement	24.87	94.28858074	0.1809679	8890.336458	8890.336458	
x4	Movement	35.99	94.30009741	0.200162343	8892.508371	8892.508371	
x5	Movement	45.28	94.32890237	0.248170611	8897.941822	8897.941822	
x6	permanent	0	94.50647904	0.544131725	8931.47458	8931.47458	

Iteration Results 1

Table 11. Iteration Results 1

POPULATION	**f(x)**	**i(i)**
x1	119.4869629	14277.13429
x2	140.4514367	19726.60606
x3	119.5405002	14289.9312
x4	130.4995765	17030.13948
x5	139.7974517	19543.3275
x6	94.77804309	8982.877452
	f(x)	**i(i)**
Global Bests	140.4514367	19726.60606

In the first iteration, the brightest is in fireflies 2.

Then the best route for ACTS-FATS is 1-2-6-5-4-3-1 and the optimum route is obtained at epoch 1 with a distance of 87.25 which can be seen in the tabulis list.

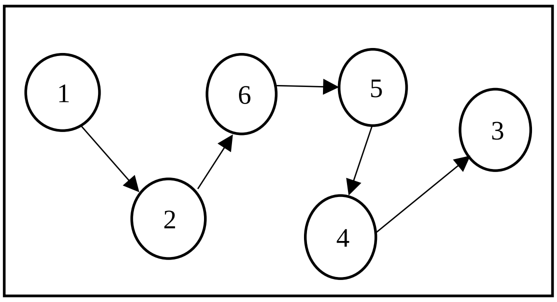

Figure 3. ACTS-FATS Best Route

DISCUSSIONS

Table 12. Results of the TSPLIB95 A280 Dataset Experiment with ACTS-FATS

No	**Iteration**	**Optimum Distance (ACTS-FATS results)**	***Running Time***
1	30	54734.7814	0.384237 seconds.
2	50	34393026	0.385471 seconds.
3	100	22202.2237	0.484937 seconds.
4	150	22202.2237	0.180539 seconds.
5	200	22202.2237	0.183859 seconds.
6	250	22170.3878	0.304696 seconds.
7	300	22105.8323	0.297748 seconds.

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Issue 1, January 2023
DOI : https://doi.org/10.33395/sinkron.v8i1.12016
e-ISSN : 2541-2019
p-ISSN : 2541-044X

8	400	22105.8323	0.179085 seconds.
9	450	22105.8323	0.175525 seconds.
10	500	22084.4767	0.176454 seconds.

In the TSPLIB95 A280 dataset, the best results for ACTS-FATS include the number of iterations of 100 and 76 populations. The results obtained in increasing the TSP solving on the TSPLIB95 A280 dataset are 22084.4767 km with running 0.176454 seconds. Route images and iterative graphs for the best results in improving TSP solutions on the TSPLIB95 A280 dataset, can be seen in Figures 4.11 and 4.12 below.

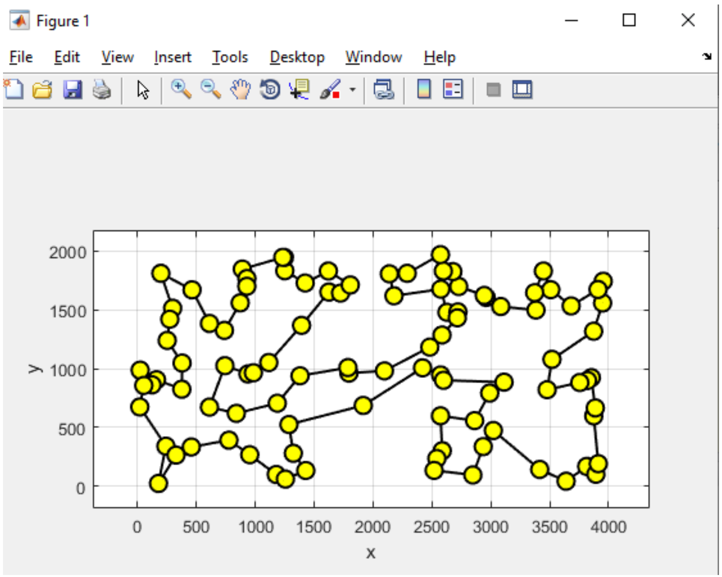

Figure 4. Best Results of ACTS-FATS on the TSPLIB95 A280 Dataset

While the graph of the best iteration results on the TSPLIB95 A280 dataset can be seen in the image below.

Table 13. Comparison Results Against Previous Research

No	Datasets	Previous Research Results	ACTS-FATS results	Difference	Best result	Enhancement accuracy
1	TSPLIB95 A280	22851.76	22084.4767	767.29	22084.4767	3.47 %
2	Eil51	429.11	366.6241	62.49	366.6241	17.0 %
3	Eil76	548.37	530.0558	18.32	530.0558	3.45 %
No	**Datasets**	**Previous Research RT Results**	**Results of RT ACTS-FATS**	**Difference**	**Best result**	**Enhancement accuracy**
1	TSPLIB95 A280	0.225498	0.176454	0.049044	0.176454	27.79 %

*name of corresponding author

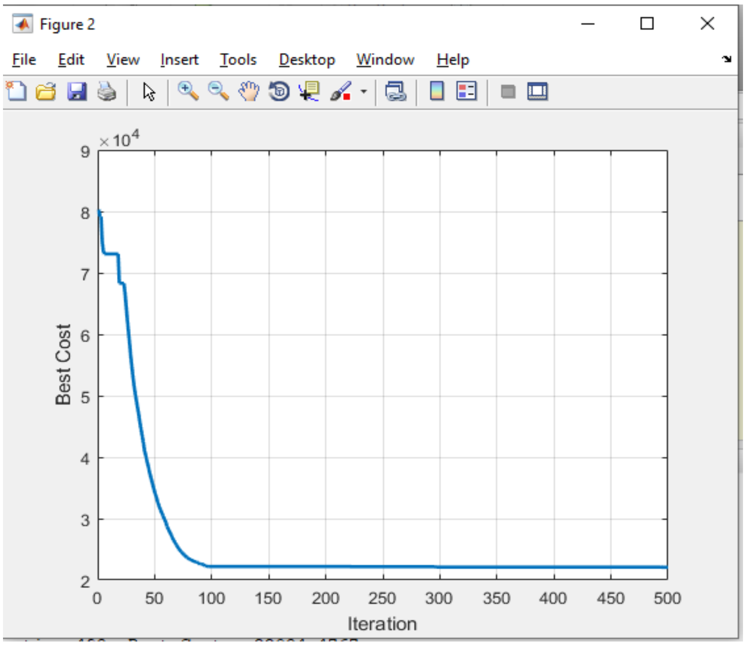

Figure 5. Graph of Iteration Results on the TSPLIB95 A280 Dataset

From the data in table 13 above, it was obtained that the accuracy increase in the Eil51 dataset was 17%, while in the Eil76 dataset it was 3.45%, and in the TSPLIB95 A280 dataset it was 3.47% with a running time of 27.79%.

CONCLUSION

From the results of the research experiment, it can be concluded that the TSP (Travelling Salesman Problem) can be optimized by combining two algorithms, namely the ACTS algorithm and the FATS algorithm. This is evidenced by the results that get better running time in solving the TSP (Travelling Talesman Problem), where the FATS algorithm functions as a controller for the selected routes and optimal results are obtained, by obtaining the percentage value for increasing accuracy in the ACTS- FATS . Experiments on the Eil51 dataset obtained an increase in accuracy of 17%, while on the Eil76 dataset it was 3.45%, and on the TSPLIB95 A280 dataset it was 3.47% with a running time of 27.79%.

REFERENCES

Alaidi, A. H. M., & Mahmood, A. (2018). Distributed hybrid method to solve multiple traveling salesman problems. *International Conference on Advances in Sustainable Engineering and Applications, ICASEA 2018 - Proceedings*, 74–78. https://doi.org/10.1109/ICASEA.2018.8370959

Alobaedy, M. M., Khalaf, A. A., & Muraina, I. D. (2017). Analysis of the number of ants in ant colony system algorithm. *2017 5th International Conference on Information and Communication Technology, ICoIC7 2017*, *0*(c), 3–7. https://doi.org/10.1109/ICoICT.2017.8074653

Deng, W., Xu, J., & Zhao, H. (2019). An Improved Ant Colony Optimization Algorithm Based on Hybrid Strategies for Scheduling Problem. *IEEE Access*, *7*, 20281–20292. https://doi.org/10.1109/ACCESS.2019.2897580

Dewantoro, R. W., Sihombing, P., & Sutarman. (2019). The Combination of Ant Colony Optimization (ACO) and Tabu Search (TS) Algorithm to Solve the Traveling Salesman Problem (TSP). *2019 3rd International Conference on Electrical, Telecommunication and Computer Engineering, ELTICOM 2019 - Proceedings*, 160–164. https://doi.org/10.1109/ELTICOM47379.2019.8943832

Gadioli, D., Nobre, R., Pinto, P., Vitali, E., Ashouri, A. H., Palermo, G., Cardoso, J., & Silvano, C. (2018). SOCRATES - A seamless online compiler and system runtime autotuning framework for energy-aware applications. *Proceedings of the 2018 Design, Automation and Test in Europe Conference and Exhibition, DATE 2018*, *2018-Janua*, 1143–1146. https://doi.org/10.23919/DATE.2018.8342183

Hartono, Mhd. Furqan, Tulus, E. B. N. (2015). *Determining Membership Function of Fuzzy Logic Using Genetic Algorithm based on Max-Min Composition*. *November*, 1–5.

Hay's, R. N. (2017a). Implementasi Firefly Algorithm-Tabu Search Untuk Penyelesaian Traveling Salesman Problem. *Jurnal Online Informatika*, *2*(1), 42. https://doi.org/10.15575/join.v2i1.63

*name of corresponding author

Hay's, R. N. (2017b). Kombinasi Firefly Algorithm-Tabu Search untuk Penyelesaian Traveling Salesman Problem. *Jurnal Online Informatika*, *2*(1), 42. https://doi.org/10.15575/join.v2i1.63

Meng, L., Lin, Y., Qing, S., & Wenjing, F. (2019). Research on Generalized Traveling Salesman Problem based on Modified Ant Colony Optimization. *Proceedings of the 31st Chinese Control and Decision Conference, CCDC 2019*, 4570–4574. https://doi.org/10.1109/CCDC.2019.8833167

Min, B., Shipin, Y., Xu, Y., & Lijuan, L. (2019). An Improved Ant Colony Algorithm for Traveling Salesman Problem. *Proceedings of 2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference, IAEAC 2019*, *Iaeac*, 1046–1050. https://doi.org/10.1109/IAEAC47372.2019.8997589

Septiyafi, I., Suprajitno, H., & Pratiwi, A. B. (2019). Penerapan Algoritma Kunang-Kunang pada Open Vehicle Routing Problem (OVRP). *Contemporary Mathematics and Applications (ConMathA)*, *1*(1), 46. https://doi.org/10.20473/conmatha.v1i1.14774

Yang, X. S. (2010). Firefly algorithm, stochastic test functions and design optimization. *International Journal of Bio-Inspired Computation*, *2*(2), 78–84. https://doi.org/10.1504/IJBIC.2010.032124

Yang, X., & Wang, J. S. (2016). Application of improved ant colony optimization algorithm on traveling salesman problem. *Proceedings of the 28th Chinese Control and Decision Conference, CCDC 2016*, 2156–2160. https://doi.org/10.1109/CCDC.2016.7531342

*name of corresponding author