Ant Colony Algorithm with Fuzzy Logic

Bulanık Mantıklı Karınca Kolonisi Algoritması

Ramazan AKKULAK, Eskişehir Osmangazi University, Electrical-Electronics Engineering

Özetçe— **Bu çalışmada küresel bir problem haline gelmiş gezgin satıcı problemine bulanık mantık kullanılarak oluşturulan algoritmadan bahsedilmektedir. Gezgin satıcı problemi; uzaklıkları bilinen noktaların her birine bir defa gidecek şekilde en az maliyetli rotayı hesaplama problemidir. Günümüzde rotalama problemi birçok uygulama alanında karşımıza çıkmaktadır. Gezgin satıcı problemine kesin ve sezgisel olarak çeşitli çözüm yolları geliştirilmekle birlikte NP-zor problemi olduğu bilinmektedir. Bu problemin çözümü Charles Darwin'in doğal seçilim teorisinden esinlenilen genetik algoritma ve karıncaların davranışlarından esinlenilen karınca koloni algoritmaları gibi metotlarla çözümlenmiştir. Çözümün sonucu genellikle gidilecek konum sayısı ve uzaklıkların yerel çözümü belirlemektedir. Geliştirilen algoritma temeli karınca kolonisi algoritması olmakla birlikte bulanık mantık sayesinde başarılı sonuçlar vermesi hedeflenmiştir. Yapılacak her hesaplama, her bir kombinasyon için farklı olduğundan en iyi çözümü veren bulanık bir sistemle desteklenmiştir. Geliştirilen bu sistem üç giriş, bir çıkış durumuna göre tasarlanmıştır. Yapılan hesaplamalar sonucunda bulanık mantıkla desteklenen karınca kolonisi algoritmasının en iyi çözümü bulduğu gözlemlenmiştir.**

Anahtar Kelimeler — gezgin satıcı problemi; bulanık mantık; karınca kolonisi; genetik.

Abstract— **In this study, the algorithm created by using fuzzy logic for the traveling salesman problem, which has become a global problem, is mentioned. Traveling salesman problem; It is the problem of calculating the least costly route to go once to each of the points whose distances are known. Today, the routing problem is encountered in many application areas. Although various precise and intuitive solutions have been developed to the traveling salesman problem, it is known that there is a NP-hard problem. The solution to this problem was solved by methods such as genetic algorithm inspired by Charles Darwin's theory of natural selection and ant colony algorithms inspired by the behavior of ants. The result of the solution usually determines the local solution of the number of destinations and distances to go. Although the basis of the developed algorithm is the ant colony algorithm, it is aimed to give successful results thanks to fuzzy logic. Since each calculation to be made is different for each combination, it is supported by a fuzzy system that gives the best solution. This developed system is designed for three inputs and one output. As a result of the calculations, it was observed that the ant colony algorithm supported by fuzzy logic found the best solution.**

Keywords — travelling salesman problem; fuzzy logic; ant colony; genetic.

I. INTRODUCTION

The routing solution of a seller to n different cities by the shortest route and on the condition of not visiting the same city again is called the traveling salesman problem. In the literature, traveling salesman problem (TSP) is known as a problem that finds the shortest path using the n-point closed-loop system given when it has to go to more than one destination. The routing problem is an important problem of cargo companies, mobile robots. Today, developing technology uses the TSP solution a lot in artificial intelligence applications and robot applications. The higher the number of points in the TSP solution, the harder it is to obtain the solution. For this reason, precise and heuristic algorithms have been developed. Since TSP is a very high combination problem, it is known that the developed algorithms provide optimization to the solution of the problem. Figure 1 shows a routing (TSP) problem and its solution.

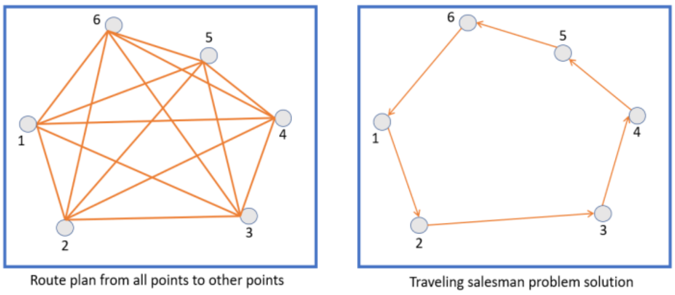

Figure 1 TSP

TSP is defined as the NP-Hard (Non-deterministic-polynomial Time - hard) problem. Precise and heuristic algorithms have been developed to the TSP solution. In the literature, we first encounter with the genetic algorithm developed using Charles Darwin's theory of evolution as the TSP solution. In the study conducted [1-2], it was emphasized that the genetic algorithm (GA) has three important steps. These are respectively, coding, fitness function, transition, and mutation. It has been stated that while GA uses a random search technique, it produces the appropriate solution thanks to the concepts of crossing and mutation, which is the theory of Charles Darwin.

The processing steps of the genetic algorithm are shown in figure 2 [3]. According to this situation, individuals are lined up from the possible solutions. While each individual is named with a chromosome, the structure that consists of chromosomes is called a population. The fitness value of each chromosome in the population is calculated. After the fitness value of the individuals is calculated, the individuals to be crossed are selected. After crossover, it enters the mutation process so that the solution does not get stuck in the local solution. The fitness values of the population are recalculated with the newly created generations.

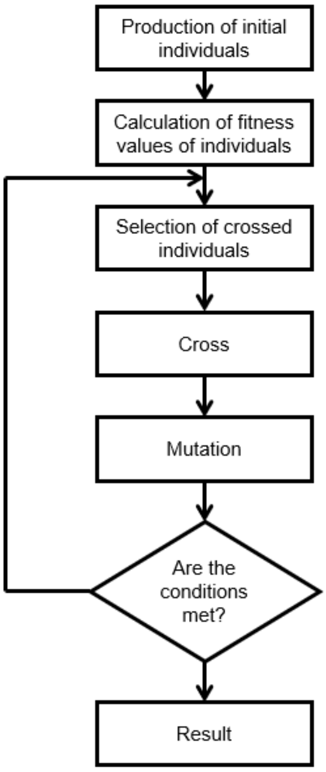

Figure 2. GA Flow Diagram

These processes are repeated continuously to find the most suitable solution. The sample population, crossover and mutation are shown in figure 3. Each number represents the location location to be entered.

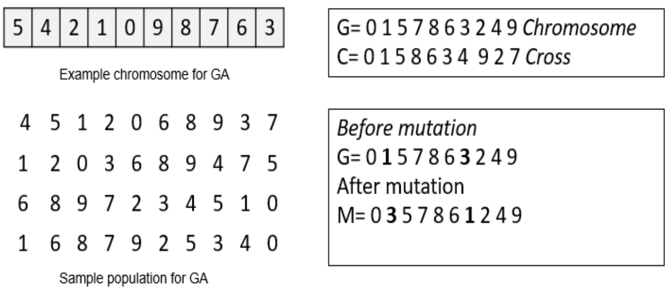

Figure 3. The sample population, crossover, and mutation

The genetic algorithm pseudocode is as follows:
Step 1: The initial population is created.
Step2: The fitness values of individuals in the population are calculated.
Step3: Individuals in the population are selected for crossover according to the individual selection method.
Step 4: Selected individuals are crossed according to the crossover probability and new individuals are created.
Step 5: New individuals are mutated according to the mutation probability.
Step 6: New individuals are identified as existing members of the population.
Step 7: Go to Step 2 until the specified number of generations is reached or the target value is reached.

Another approach to the TSP solution is the ant colony algorithm inspired by ant behavior. It was first handled by Dorigo and Gambardella [4]. In the study, the natural behavior of ant colonies when they find their food was mentioned. It has produced an intuitive solution for TSP. Ants secrete a hormone called pheromone and perform the nest-food relationship by doing this shortest routing [5]. Nest-food relationship was associated with TSP and the problem of routing with the pheromone hormone coefficient value was solved. It has been observed that the ant colony algorithm (ACO) is successful in TSP solution of large amount of target locations.

Ant nest-food Relationship can be defined as follows; When an ant finds a food source, it evaluates the quality and quantity of the source. Then the ant takes some of the food and returns to its nest. When the ant returns, it secretes the pheromone hormone in direct proportion to these two characteristics of the food and spreads it on the path it passes. Thus, other ants perceive the quality and quantity of the food at the end of this path thanks to the amount of this hormone. In some animal species such as insects and reptiles, pheromone is the type of hormone they use to impress their mates of their own kind, to convey a warning or to trace. Ants also use this hormone to find the shortest path to food. As ants move forward, they store a certain amount of pheromone. The density of the pheromone material will be higher at the shortest distance between the nest and the food. It has been observed that the path with the highest pheromone density is the best path.

When an obstacle occurs on the way between the nest and the food (Figure 4), the ant that comes in front of the obstacle has to make a choice. This way there is a chance to choose right or left. Ants go the way they choose. The new paths formed here begin to be preferred by the ants coming from behind after a certain period of time, and the most preferred path to the result of this situation, that is, the path with the most pheromone material becomes the shortest path.

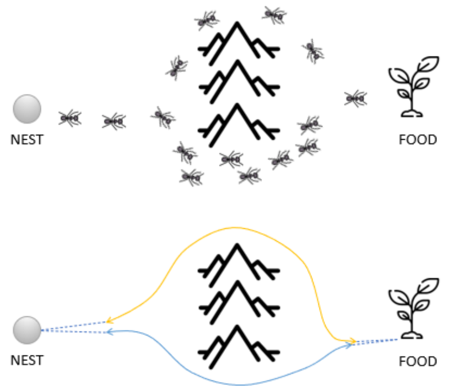

Figure 4. Ant's nest-food relationship

Based on the developed algorithm, there is the creation of an ant tour. First, it is determined how many ants there will be in a colony. Then, each ant is placed in a random spot and these ants visit the nodes one by one and complete their tour. Ants leave pheromones on their way to each destination they visit. Routes are calculated for each ant. This value is called the local solution. Then, the pheromones evaporate due to the structure and application of the pheromone they secrete. The pheromones formed on some routes lose their value by flying. This situation is repeated continuously until the desired number of ants is reached. The flow chart is shown in Figure 5 below.

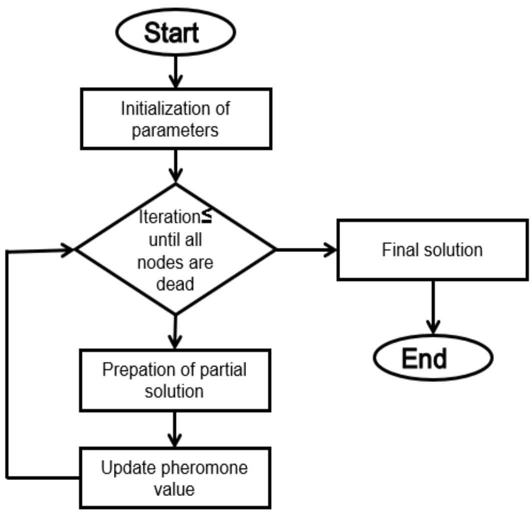

Figure 5. ACO flow chart

The ant colony algorithm pseudocode is as follows:
Step 1: Ants are created and randomly placed in cities.
Step 2: Each ant creates routes by roaming the cities according to the specified alpha and beta values.
Step 3: The route distance of each ant is calculated.
Step 4: The pheromone values on the roads are increased according to the route distances of the ants.
Step 5: According to the pheromone evaporation value, the pheromone value in all paths is reduced.
Step 6: Go to Step 2 until the specified number of generations is reached or the target value is reached.

The TSP solution is very important for mobile robots, which are widely used today. Since mobile robots have energy states, they need to use their energy in the shortest way and in the most efficient way. The ACO and GA methods we mentioned earlier provide solutions to the Traveling Salesman problem. We developed an ant colony algorithm working with fuzzy logic in our own project. Thanks to this algorithm, a mobile robot to be developed can perform the given tasks by using the shortest routing and using its energy in the most efficient way. The designed fuzzy logic algorithm mathematical algorithms are given in equation 1,2 and 3.

The distance equation formula is the basis of the Traveling Salesman problem. When the x and y values of the points in two different positions are known, the distance between them is found by the equation 1.

$$Distance = \sqrt{(\mathrm{x}_1 - \mathrm{x}_0)^2 + (y_1 - y_0)^2} \qquad (1)$$

If there is more than one position in the TSP, different routing situations that may occur are found with the help of permutation. It helps to find out how many different routing situations can occur in Equation 2. The "n" is represents the total destinations to go.

$$Roution\ Statues\ = n\,! \qquad (2)$$

There are *n* different positions to start with. First position, the problem (n – 1) has a choice between different positions. Second position, the problem has a choice between *(n – 2)* different positions. Third position, the problem has a choice between *(n – 3)* different positions. It continues until it has a final position. Ant colony general mathematical formula is given in formula 3. Calculations are made with the help of the formulas given in equation 1 and equation 2 and the shortest solution path is found.

$$p_{ij}^{k} = \begin{cases} \frac{\tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}}{\sum_{l \in N_k(i)} \tau_{il}^{\alpha} \cdot \eta_{il}^{\beta}} & \text{if } j \in N_k(i), \\ 0 & \text{otherwise.} \end{cases} \qquad (3)$$

α : pheromone coefficient

β : intuitive coefficient

N: set of nodes

η_{ij}=heuristic value between nodes i and j

τ_{ij}=The pheromone value between nodes i and j.

Heuristic approaches are generally available in the algorithms used in solving the TSP problem. Combining these approaches with fuzzy logic, it is possible to produce faster and more accurate results. Thanks to the formulas given above, TSP solution is solved with fuzzy logic ant colony.

II. FUZZY LOGIC MODEL

The developed algorithm is supported by fuzzy logic. The fuzzy membership functions of the algorithm we will use in the TSP solution are determined as colony number, iteration and pheromone density. Fuzzy logic membership functions are given in Figure 6.

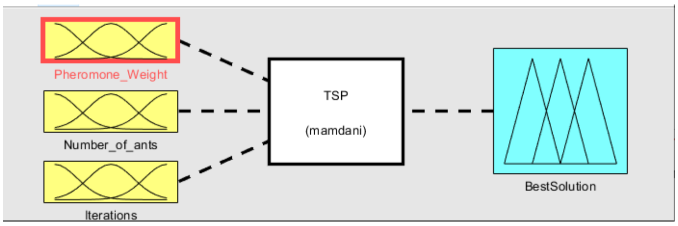

Figure 6. Fuzzy inference system

The fuzzy inference system is mamdani model and consists of 3 inputs and 1 output. The input memberships of the fuzzy system are shown in figure 7.

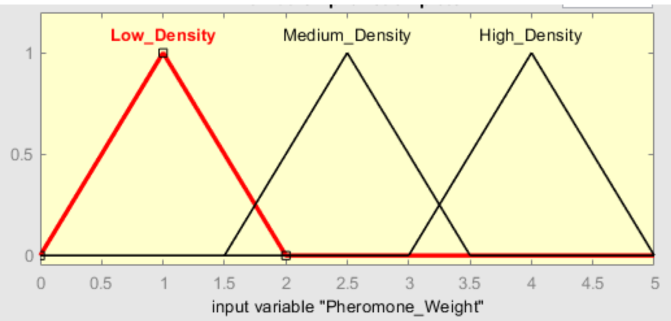

Figure 7.a Pheromone Weight membership

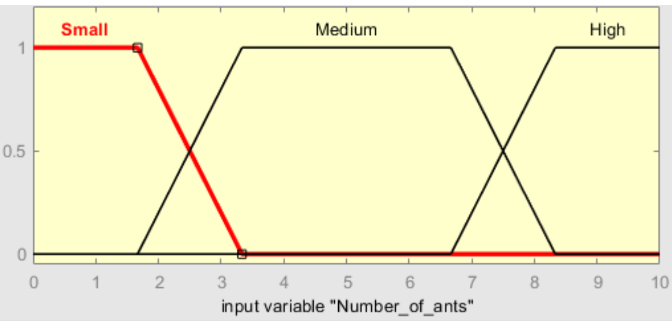

Figure 7.b Number of ants membership

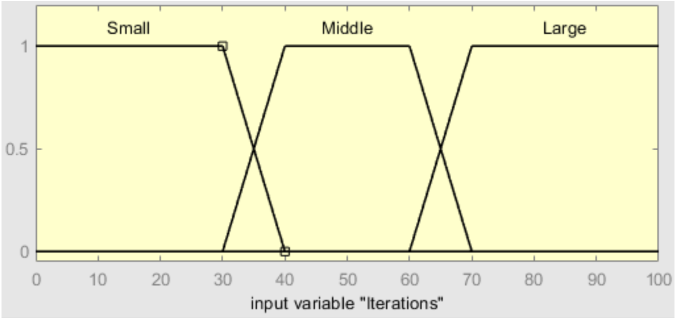

Figure 7.c Iterations membership

The output membership, which shows that the routing obtained in the TSP solution is long or short, within the rules determined according to the given membership of output functions, is shown in figure 8.

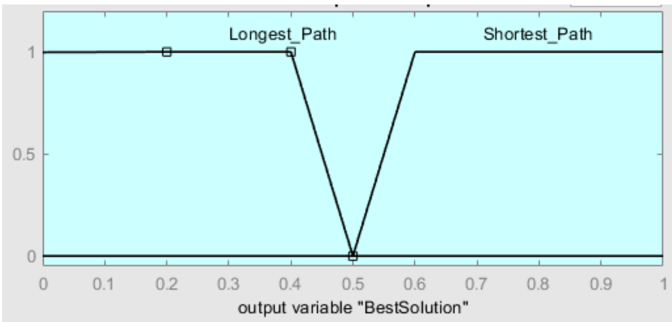

Figure 8. Best solutions membership

There are two different cases to observe the effect of the input memberships of the fuzzy logic system designed for the TSP on the output. These two situations are: determined as long path and short path. The algorithm developed according to the given input dependencies determines the length of the selected path.

The list of rules that will occur as a result of the entry and exit of membership functions is given in the table 1.

TABLO 1: FUZZY RULES

Rule 1. If (Pheromone Weight is high density) and (Number of ants is high) and (Iterations is Large) then (Best solution is shortest path).
Rule 2. Rule 1. If (Pheromone Weight is high density) and (Number of ants is medium) and (Iterations is Large) then (Best solution is shortest path).
Rule 3. Rule 1. If (Pheromone Weight is high density) and (Number of ants is small) and (Iterations is Large) then (Best solution is shortest path).
Rule 4. Rule 1. If (Pheromone Weight is high density) and (Number of ants is high) and (Iterations is Middle) then (Best solution is shortest path).
Rule 5. Rule 1. If (Pheromone Weight is high density) and (Number of ants is medium) and (Iterations is Middle) then (Best solution is shortest path).

Rule 6. If (Pheromone Weight is high density) and (Number of ants is small) and (Iterations is middle) then (Best solution is longest path).
Rule 7. If (Pheromone Weight is high density) and (Number of ants is small) and (Iterations is small) then (Best solution is longest path).
Rule 8. If (Pheromone Weight is high density) and (Number of ants is medium) and (Iterations is small) then (Best solution is longest path).
Rule 9. If (Pheromone Weight is high density) and (Number of ants is high) and (Iterations is small) then (Best solution is shortest path).

The algorithm has 27 different rules in total. In this table, we show 9 different situations. The ant pheromone density was chosen as high and the routing conditions were given as two different examples according to the number of repetitions and changes in the number of ants, figures 9 and 10.

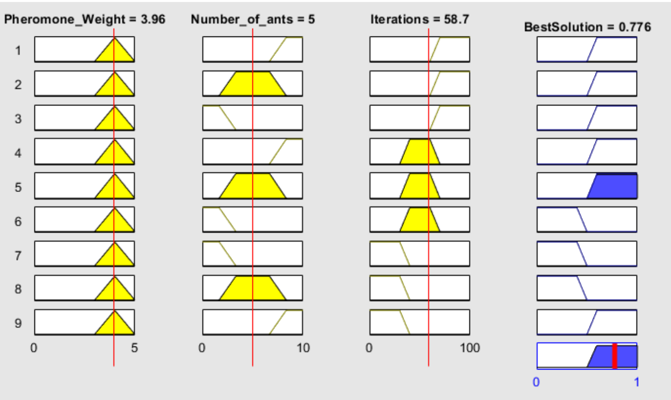

Figure 9. Example of output, Shortest path

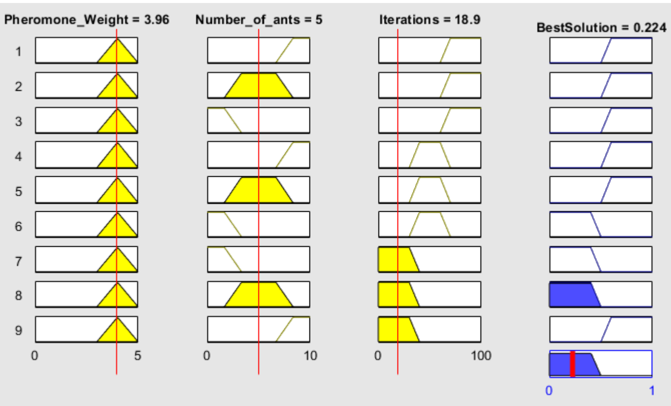

Figure 10. Example of output, Longest path

The developed algorithm is based on fuzzy logic. The algorithm developed with the inputs and outputs of the fuzzy system gives the same results. For example, when the number of iterations is low, the algorithm may not be able to reach the number of paths that will occur in line with the combination results. In this case, it is expected that the routing will be long in line with both the algorithm and the developed fuzzy logic. Although the algorithm developed in general changes depending on the number of iterations and the number of ants, the routing can also change with the changes of the special value such as pheromone weight, pheromone evaporation rate, and beta.

III. Application

The developed algorithm was developed on the python platform. The outputs of the algorithm supported by fuzzy logic were formed to the TSP solution as shown in figure 11.

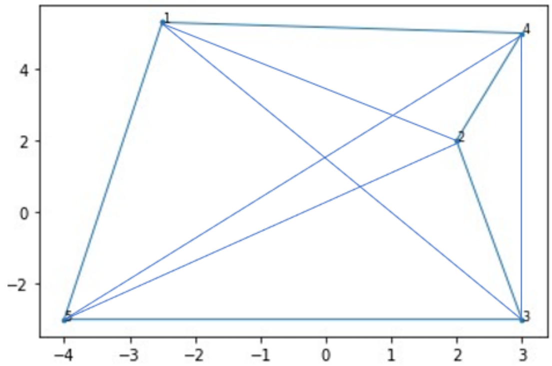

Figure 11 a. Before TSP

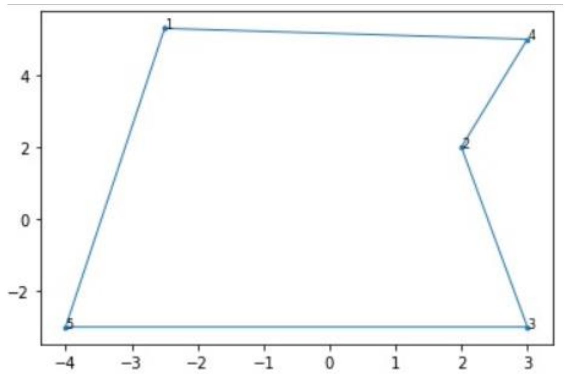

Figure 11 b. After TSP

The basis of the TSP algorithm, which is supported by fuzzy, is based on the ant colony. In general, it chooses the most appropriate path for TSP in line with the rules to be calculated.

The application of the developed algorithm on Robot Operating System (ROS) will be examined. The application tried on Turtlebot3 gave successful results. We operate our primary robot in the gazebo environment. Since we will perform robot movements via *move_base [6,7]*, we run simultaneous localization and Mapping[7] (SLAM). Before performing these steps, a media scan is required. It is shown in figure 12 that the robot is operated on the environment in gazebo.

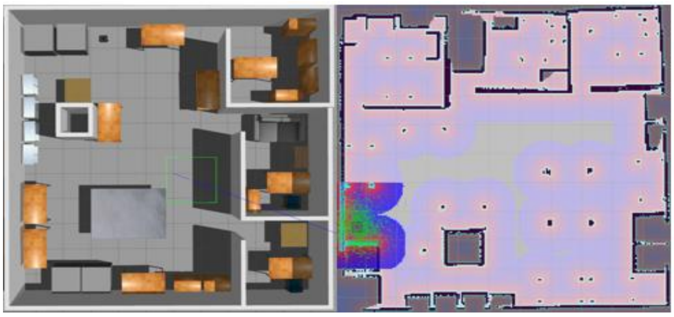

Figure 12. Robot in Gazebo software

After running the robot in the gazebo environment, move_base needs to be run. In figure 12, the image of the robot in SLAM and move_base environment is given.

Then we activate our python code and enter the locations where the robot will go. We enter the locations where the robot will go in order. The first point we enter must be the starting position of the robot. Because the robot is already in a certain position at the beginning, this position must be included in the TSP solution. It should come to this location last due to algorithm solution. The points entered in the coordinate plane are as follows: (-2.5, 5.3), (3,5), (2,2), (3,-3), (4,-3). The locations where the robot will go after these given location points are shown in figure 13.

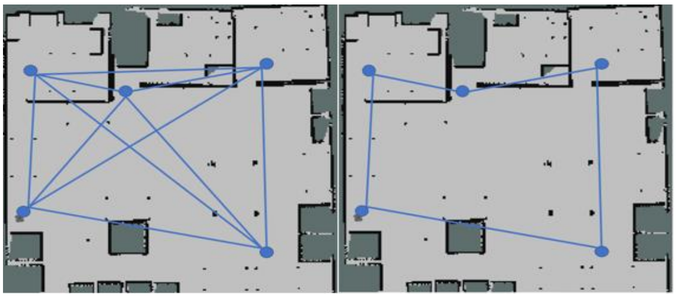

Figure 13. Before TSP (left), after TSP (left)

The results of the visit of the robot are given in figure 13, and 14 as examples.

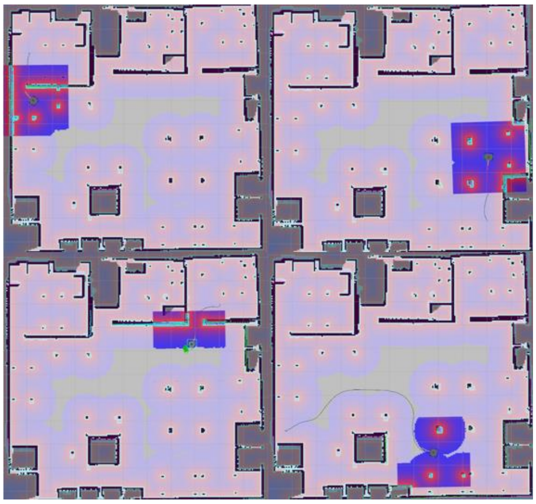

Figure 14. The movements of the robot are shown

The robot has successfully completed its routing in the ROS environment. It has been observed that the ant colony algorithm developed with fuzzy logic works more successfully than heuristic algorithms. The demo of ACO supported by fuzzy logic is given in reference 11. [11]

IV. Conclusion

Today, people face the problem of routing in many different areas. Solutions to this problem have been obtained using different methods that are precise and intuitive. In general, how a TSP approach is made is expressed in this context, and the solution approach logic of an algorithm used in solving a TSP problem is tried to be expressed.In this study, fuzzy logic inference of a TSP was performed and an ant colony algorithm with fuzzy logic was developed. In line with the results, it was tried to decide how to choose the appropriate orientation. With the algorithm to be developed in line with these results, it has been possible to develop an algorithm that gives faster results for the TSP solution. Today, it is used in many areas where TSP solution is needed. Since it is known that mobile robots are rapidly becoming widespread in the global world, TSP solution is gaining importance for robots used in fields such as multitasking and post. The combination of the developed algorithm and fuzzy system is aimed to optimize the robots used in these areas.

References

[1] Holland, J. H. (1992). Genetic algorithms. Scientific american, 267(1), 66-73.

[2] Dwivedi, V., Chauhan, T., Saxena, S., & Agrawal, P. (2012). Travelling salesman problem using genetic algorithm. *IJCA Proceedings on Development of Reliable Information Systems, Techniques and Related Issues (DRISTI 2012)*, *1*, 25.

[3] Dorigo, M., Birattari, M., & Stutzle, T. (2006). Ant colony optimization. *IEEE computational intelligence magazine*, *1*(4), 28-39.

[4] Dorigo, M., & Gambardella, L. M. (1997). Ant colonies for the travelling salesman problem. *biosystems*, *43*(2), 73-81.

[5] Yen, C. T., & Cheng, M. F. (2018). A study of fuzzy control with ant colony algorithm used in mobile robot for shortest path planning and obstacle avoidance. *Microsystem Technologies*, *24*(1), 125-135.

[6] move_base, Mevcut: http://wiki.ros.org/move_base, Date of access: 08.06.2021

[7] Leonard, J. J., & Durrant-Whyte, H. F. (1991). Simultaneous map building and localization for an autonomous mobile robot. In IROS (Vol. 3, pp. 1442-1447).

[8] navigation, Mevcut: http:// wiki.ros.org/navigation, Date of access: 08.06.2021

[9] Li, X., & Choi, B. J. (2013). Design of obstacle avoidance system for mobile robot using fuzzy logic systems. *International Journal of Smart Home*, *7*(3), 321-328.

[10] Mitchell, S., Ernest, N. D., & Cohen, K. (2013). Comparison of fuzzy optimization and genetic fuzzy methods in solving a modified traveling salesman problem. In *AIAA Infotech@ Aerospace (I@ A) Conference* (p. 4664).

[11] Demo: https://youtu.be/g_rCZfKDDKs