Alfarama Journal of Basic & Applied Sciences

Faculty of Science Port Said University

https://ajbas.journals.ekb.eg
ajbas@sci.psu.edu.eg

http://sci.psu.edu.eg/en/

DOI:10.21608/AJBAS.2021.91699.1063

ISSN 2682-275X

January 2022, Volume 3 Issue I

Submitted: 19-08-2021
Accepted: 07-09-2021

Pages: 88-104

Solving the Shortest Path Problem by Fuzzy Ant Colony Optimization Algorithm

Eman Yousif [1,*]; Ahmed Salama [1]; M. Elsayed Wahed [2]

[1]Department of Mathematics, Faculty of Sience, Port Said University; Port Said, Egypt,

[2]Faculty Of Computers and Informatics, Suez Canal University, Ismailia ,Egypt,

***Corresponding author**: emanyousif79@hotmail.com

ABSTRACT

I will offer a new strategy for addressing the shortest path problem in two methods in this paper. For the shortest path issue in an uncertain environment, the first technique is the direct graph, and the second method is the indirect graph, and this method is compromised by the Fuzzy Dijkstra algorithm [10]. I'll start by giving a quick overview of fuzzy ant colony methods in general. Eventually, I'll discuss the benefits and drawbacks of employing fuzzy and ant colony algorithms to solve the shortest path problem, as well as my thoughts on the solutions' utility and the future of this field of computer science. The Dijkstra algorithm is a popular method for solving the shortest route problem (SPP). In this research, we use fuzzy ant colony techniques SPP in an uncertain environment and use the Fuzzy Dijkstra algorithm to compromise the output. One is figuring out how to add two edges together. The other problem is determining how to compare the distance between two pathways whose edge lengths are represented by fuzzy integers and fuzzy ant colony techniques. Two numerical examples of a transportation network are utilized to tackle these challenges and demonstrate the efficacy of the proposed strategy.

Keywords:

fuzzy programming; ant colony algorithms; shortest path; optimization problems.

1. INTRODUCTION

The shortest path problem (SPP) is classic in the computer science community. It has been studied by many people but the current standard is Djikstra's shortest path algorithm, which utilizes dynamic programming to solve the problem.

Difficult combinatorial optimization problems can be solved by the natureinspired technique called ACO [22] in a moderate amount of computation time [16]. ACO simulates the behavior of ant colonies in identifying the most efficient routes from their nests to food sources [17]. Dorgio initiated the idea of identifying good solutions to combinatorial optimization problems by imitating the behavior of ants [19]. Ants when searching for food initially search the region adjacent to their nest in a random manner. The ant estimates the quantity and the quality of the food as soon as the food source is identified and takes a portion of it back to the nest [20]. An aromatic essence termed as pheromone is used by real ants to communicate with each other [23]. A moving ant marks the path by a succession of pheromone by laying some of this substance on the ground [21]. Both the length of the paths and the quality of the located food source determine the quantity of the pheromone dropped on the paths [23]. The pheromones have to evaporate for a longer time if it takes more time for the ant to travel to its nest and return again [15]. The reason for the ants to select the shorter path has been found to be due to the pheromone concentration deposited mostly on such paths [24]. Therefore, computational problems which can be downsized to determination of good paths can be solved by ACO through graphs by using the concept of "ants" [6]. By employing this concept, a population of artificial ants that searches for optimal solutions is created by the combinatorial optimization problem creates according to the constraints of the problem [25]. Combinatorial optimization problems such as Routing problem (e.g., Traveling Salesman Problem (TSP) and Vehicle Routing Problem (VRP)), Assignment problem (e.g., Quadratic Assignment Problem), Scheduling problem (e.g., Job Shop) and Subset problem (e.g., Multiple Knapsack, Max Independent Set) extensively employ the ACO algorithms [14, 19]. A majority of these applications require exponential time in the worst case to determine the optimal solution as they belong to NP-hard problems [27] [20]. In VRP the vehicle returns to the same city from where it started after visiting several cities [4].

Essentially what the shortest path problem deals with is if you have a graph G = (N,V); where N is a set of nodes or locations and V is a set of vertices that connect nodes in N where V is a subset of NxN. In SPP each vertex in V also has a weight associated with it and the problem that needs to be solved it how to get from any node in N to any other node in N with the lowest weight on the vertices used.

A simple example is to let N be all the airports serviced by an airline, and V is the flights for that airline. Additionally let the weight of V be the cost of each flight. The problem to be solved in this situation would be to find the cheapest way to get from any airport serviced to any other airport.

2. Related Works :

Some of the recent research works related to ant colony optimization are discussed below. Chen et al. [22] have introduced a two-stage solution construction rule possessing two-stage ACO algorithm to solve the large scale vehicle routing problem. Bin et al. [23] have proposed an enhanced ACO to solve VRP by incorporating a new strategy called ant-weight strategy to correct the increased pheromone, and a mutation operation. Tao et al. [24] have proposed a fuzzy mechanism and a fuzzy probable mechanism incorporating unique fuzzy ACS for parameter determination. Chang et al. [17] have proposed an advanced ACO algorithm to improve the execution of global optimum search. Berbaoui et al. [18] have proposed the optimization of FLC (Fuzzy Logic Controller) parameters of SAPF through the application of the ACO algorithm. Gasbaoui et al. [14] have proposed an intelligent fuzzy-ant approach for the identification of critical buses and optimal location and size of capacitor banks in electrical distribution systems. Salehinejad et al. [18] have discussed a multi parameter route selection system employing an ACS, where the local pheromone has been updated using FL for detecting the optimum multi parameter direction between two desired points, origin and destination. Djikstra developed an algorithm for this that can work out the problem that runs in O(n log n) time. However, it needs to be recalculated every time there is a change. The goal is to find an algorithm that can adapt to a changing graph topology in semi-real time.

Dubois discussed [1] the problem of the shortest hazy road for the first time in 1980, noting that while the shortest distance may be established, the shortest path cannot. [2] Klein also provided a hazy algorithm for each arc length within an integer range of one to a set number dependent on specialized programming. Liu also presented the [6] algorithm for determining the shortest path, although it was not prominent. As a result, they must use the Yager classification method [8] to turn all ambiguous arcs into clear brackets, solving the problem of the shortest strange path with clear arcs. In their technique, however, 0-1 variables are required. Okada [7] explored the Fuzzy Shortest Path technique by altering the numerous labeling methods for the classic Shortest Path algorithm for Fuzzy graphs with L-R type Fuzzy numbers as link weights.

Lin [5] calculated the Fuzzy Shortest Path and the most vital arc by using a Fuzzy linear programming approach. Kung [4] proposed Fuzzy Shortest Length Heuristic Procedure where the Shortest Path was decided by examining the Intersection Area (IA). In the case of the Triangular Fuzzy Number, however, the Fuzzy Similarity Degree was used to determine the shortest path. The fuzzy shortest length [2, 5] is an extension of this.

If the length of the path in the network is a trapezoidal fuzzy set, and the highest Similarity Degree is used to find the Shortest Path, a heuristic technique is presented in this study (SP). To show our recommended strategy, we've given several examples.

ACO is a probabilistic method for resolving complicated computing problems, such as determining the best route through a network. The ant colony method is grounded on the execution of ants seeking for a path between their colony and a source of food to identify the best path in a graph. It also appeared intriguing to address complex computation issues with something as simple as ants. These innovative methods can be utilized to cut down on the amount of time it takes to solve telecommunications optimization problems.

The purpose of using the ant colony approach is to find the best path in a graph grounded on the execution of ants searching for a way to get from their colony to a food source. It also seems to be intriguing to employ something as basic as ants to solve complex calculation issues. These one-of-a-kind solutions are used to solve telecommunication optimization problems in less time [2].

ACO (ant colony optimization) [1,3] is an algorithm based on real ant behavior in determining the shortest path from a source to food [1]. The best path from source to destination is determined by an ant colony. To make better use of ACO, this work suggests routing ant colony optimization to choose the best path based on the least latency (from source to destination) and maximum bandwidth (for each connection) for solving routing problems.

This work proposes routing ant colony optimization to select optimal path based on least delay (delay from source to destination) and maximum bandwidth (for each link) for solving routing problem in order to make better use of ACO. The computer network which is considered in this work is modeled as graph shown in figure 4. Figure 5 shows connection matrix of network in figure 4.

We use of fuzzy programming technique to fuzzy random multi-objective unbalanced transportation data problems when the sources and destination parameters are fuzzy random variables in inequality type of constraints. [11], first convert the fuzzy random multi-objective unbalanced transportation problem into deterministic problem by using fuzzy random chance constraints approach. By introducing the concept of linear membership function of fuzzy programming, multi-objective deterministic transportation problem is converted into single objective deterministic problem and then we solve it and we obtain the optimal compromise solution. Lastly a numerical example is provided for illustration the methodology. [12]

3. Fuzzy numbers

Definition 2.1. “Fuzzy set”: Let X be a discourse universe. Where $\tilde{A}$ is a fuzzy subset of X; and for all x ∈X, there is a number $\mu_{\tilde{A}}$ (x) ∈ [0, 1] which represents the degree of x in $\tilde{A}$, and is termed the membership function of $\tilde{A}$ [10].

Definition 2.2. “Fuzzy number”: A fuzzy number $\tilde{A}$ is an ordinary and convex fuzzy subset of X [10]. Here, “normality” implies that: ∃x ∈ R, $\overset{\forall}{x}\mu_{\tilde{A}}$ =1, and “convex” means that:

$\forall x1 \in X, x2 \in X, \forall \alpha \in [0, 1], \mu_{\tilde{A}}(\alpha\, x1 + (1-\alpha)x2) \geq \min(\mu_{\tilde{A}}(x1), \mu_{\tilde{A}}(x2))$.

Definition 2.3. A triangular fuzzy number $\widetilde{A}$ is defined by a triplet (a, b, c), where the membership can be defined as follows. A triangular fuzzy number $\widetilde{A}$ = (a, b, c) can be presented in Fig. 1.

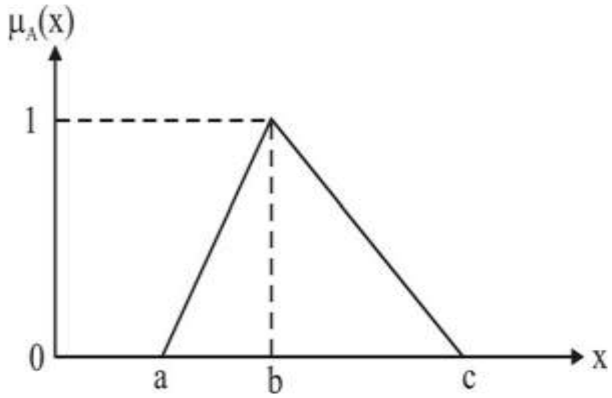


Fig. 1: Triangular fuzzy number

$$\mu_{\widetilde{A}}(x) = \begin{cases} 0, & x < a \\ \dfrac{x-a}{b-a}, & a \le x \le b \\ \dfrac{c-x}{c-b}, & b \le x \le c \\ 0, & x > c \end{cases}$$

Definition 2.4. A trapezoidal fuzzy number $\widetilde{A}$ can be defined as $\widetilde{A}$ = (a1, a2, a3, a4), where the membership can be determined as follows and shown in Fig. 2:

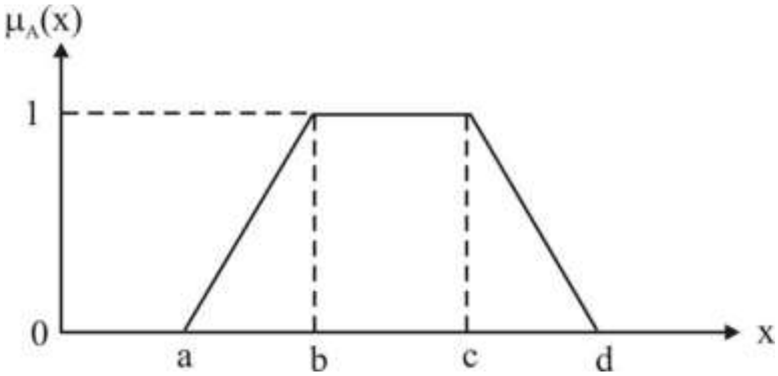


Fig.2: Trapezoidal fuzzy number

$$\mu_{\widetilde{A}} = \begin{cases} 0 & x \le a_1 \\ \dfrac{x-a_1}{a_2-a_1} & a_1 \le x \le a_2 \\ 1 & a_2 \le x \le a_3 \\ \dfrac{a_4-x}{a_4-a_3} & a_3 \le x \le a_4 \\ 0 & a_4 \le x \end{cases}$$

Definition 2.5. The graded mean integration representation of a triangular fuzzy number $\widetilde{A}$ = (a1, a2, a3) is defined as:

$$P(\widetilde{A}) = 1/6(a_1+4\,a_2+a_3) \quad (1)$$

Let's use the triangular fuzzy numbers $\widetilde{A}$= (a1, a2, a3) and $\widetilde{B}$= (b1, b2, b3). The graded mean integration representation of triangular fuzzy numbers can be derived, respectively, by applying Eq. (1):

$$P(\widetilde{A}) = \frac{1}{6}\,(a_1, a_2, a_3 \quad (2)$$

$$P(\widetilde{B}) = \frac{1}{6}\,(b_1, b_2, b_3) \quad (3)$$

Definition 2.6. The addition process $\oplus$ on triangular fuzzy numbers $\widetilde{A}$ and $\widetilde{B}$ can be simply defined as: $P(\widetilde{A} \oplus \widetilde{B}) = P(\widetilde{A}) + P(\widetilde{B}) = = \frac{1}{6}(a1+4\,a2+ a3) \; + \frac{1}{6}(b1+4b2+ b3)$ (4)

The following is the canonical representation of the multiplication operation on triangular fuzzy numbers:

$$P(\tilde{A} \otimes \tilde{B}) = P(\tilde{A}) \times P(\tilde{B}) = \frac{1}{6}(a_1 + 4 \times a_2 + a_3) \times \frac{1}{6}(b_1 + 4 \times b_2 + b_3)$$

Definition 2.7 The graded mean integration representation of a triangular fuzzy number given a trapezoidal fuzzy number $\widetilde{A}$ = (a1, a2, a3, a4) is defined as follows:

$$P(\tilde{A}) = \frac{1}{6}(a_1 + 2 \times a_2 + 2 \times a_3 + a_4) \quad (5)$$

The graded mean integration representation of operations on trapezoidal fuzzy numbers can be obtained similarly to that of triangular fuzzy numbers. Suppose that we have two trapezoidal fuzzy numbers $\widetilde{A}$= (a1, a2, a3, a4) and $\widetilde{B}$= (b1, b2, b3, b4). The graded mean integration representation of the addition process of trapezoidal fuzzy numbers can be stated as using Eq. (5) as follows:

$$p(\tilde{A} \oplus \tilde{B}) = \frac{1}{6}\,(a_1 + 2a_2 + 2a_3 + a_4) + \frac{1}{6}\,(b_1 + 2b_2 + 2b_3 + b_4)$$

The action of multiplication on trapezoidal fuzzy integers is defined as

$$p(\tilde{A} \otimes \tilde{B}) = \frac{1}{6}\,(a_1 + 2a_2 + 2a_3 + \; a_4) + \frac{1}{6}\,(b_1 + 2b_2 + 2b_3 + b_4) \quad (6)$$

4. Ant Colony Fuzzy Optimization [1-2].

We can use an ant colony algorithm to estimate the shortest path in this segment. By studying a set of linkages, we can also find nodes capable of coding. Assume that the amount of pheromone diffused through this link l is the cost way between the two items (i, j). As problem-solving continues, pheromone information is modified at random to reflect the ants' experience. As a result, the ants choose paths at random grounded on the pheromone concentration. The magnitude of the pheromone deposited on the way is similar to the path's quality, and the size of the pheromone deposited on the track is used to determine the track's quality. Since the quality here stands for the length of this way, the shortest pathway is estimated to be the largest of the pheromones on this path, L, one of the candidates for Ant Improvement.

The size of the pheromone that is deposited on the path is determined by the path's quality. The quality of a road is determined by its length; more pheromone is located on the shortest paths, L, which became Ant Colony optimization paths.

$T_i j^k(t)$ can be defined as the pheromone size in (L) connecting node i in addition to node j. An ant can move from one intermediate node *i* to the second node *j* in each iteration of the Ant Colony algorithm.

This indicates that the intermediate node has completed the intermediate solution. Furthermore, one ant (m) can calculate a set L of possible paths in each iteration and goes to one of them with a contingency of going p_{ij}^k.

The attraction (heuristic value) and trail level are two criteria that influence the likelihood of relocating. The first element can be calculated using heuristic approaches that show the action's a priori desirability, whereas the second factor represents a historical account of making similar movements in the past. That is to say, it acts as a secondary indicator of whether or not the motion is desirable. When all ants completed a list of their solution, the trails are updated on a regular basis.

It's also repressively indexed as rising or decreasing to indicate good and low quality solutions. [17] gives the chance that ant goes from node *i* to node j at time t, indicated as P:

$$\rho_{ij} = \begin{cases} \dfrac{T_{ij}^{\alpha}(t) n_{ij}^{\beta}}{\sum_{j \in N_i^k} T_{ij}^{\alpha} n_{ij}^{\beta}(t)} & j \in N_i^k \\ 0 & O/W \end{cases} \quad (7)$$

Where N_i^k is a set of nodes for ants, and m. α, 0≤ α ≤ 1 and β are two highly positive parameters that affect the relative weight of the pheromone trail and its heuristic value. The equation [8] calculates the historical value as follows:

$$n_{ij}^{\beta} = \frac{1}{d_{ij}}$$

Where d_{ij} denotes the distance that is utilized to lessen the likelihood of choosing a long path. When an ant enters the erroneous N_i^k destination, it stops travelling down the path. The set of all neighbor nodes of *i* and n_{ij} contains heuristic information. The amount of pheromone in all of the links can be determined using the formula in [8] below after all of the ants have completed one iteration:

$$T_{ij}(t) = (1-\rho) * T_{ij}(t-1) + \rho \sum_{k=1}^{m} \Delta T_{ij}^k (t-1) \quad (8)$$

After modernization, $T_{ij}(t)$ is the pheromone on the verge (*i, j*). The pheromone before modernization is $T_{ij}(t-1)$, and the number of routes in the solution is k. The vaporization degree of the pheromone is ρ ∈ {0,1}; the magnitude of the pheromone spread is ΔT_{ij}^k that may be measured using [8]:

$$\Delta T_{ij}^{k} = \begin{cases} \frac{1}{L^{k}} & if \text{ ant k use curve (i, j)} \in \text{iteration} \\ 0 & O/W \end{cases} \quad (9)$$

Where L^k is the iteration length that determined by ant k^{th} route. E^{lk} here is essential in determining the quality of each ant's solution.

This is a triangular fuzzy number. The membership function of a triangular fuzzy number $\tilde{A}$can then be defined by a triplet.
Step1: Input: $L_i = (a_i', b_i', c_i')$, i = 1,2,…,n where L_i the trapezoidal fuzzy Then $p(\tilde{A}) = \frac{1}{6}(a + 4b + c)$ If $\tilde{A}$ = (a, b, c) and $\tilde{B}$ = (e, f, g) be two triangular fuzzy numbers Then P($\tilde{A}$) = $\frac{1}{6}$ ($a_1 + 4a_2 + a_3$) , and $p(\tilde{B}) = \frac{1}{6}(e + 4f + g)$ Addition operation $p(\tilde{A} \oplus \tilde{B}) = \frac{1}{6}(a + 4b + c) + \frac{1}{6}(e + 4f + g)$ and multiplication operation $p(\tilde{A} \otimes \tilde{B}) = \frac{1}{6}(a + 4b + c)\ x\ \frac{1}{6}(e + 4f + g)$ **OR**
Step 1: (Trapezoidal **Fuzzy Number).** Let trapezoidal fuzzy number $\tilde{A}$ be defined by (a, b, c, d) and $\tilde{B}$ be defined by (e,f,g,h) the membership function then
$p(\tilde{A}) = \frac{1}{6}(a + 2b + 2c + d)$ And $p(\tilde{B}) = \frac{1}{6}(e + 2f + 2g + h)$ The addition process of trapezoidal fuzzy numbers $\tilde{A}$ and $\tilde{B}$ can be defined as: $p(\tilde{A} \oplus \tilde{B}) = \frac{1}{6}(a + 2b + 2c + d) + \frac{1}{6}(e + 2f + 2g + h)$ And the multiplication process on trapezoidal fuzzy numbers $\tilde{A}$ and $\tilde{B}$ is defined as $p(\tilde{A} \otimes \tilde{B}) = \frac{1}{6}(a + 2b + 2c + d)\ x\ \frac{1}{6}(e + 2f + 2g + h)$
Step 2: The table of pheromone plays the part of the routing tables where the ways are implicitly determined by the values of the variables of pheromone found in this table - The minimum delay and the bandwidth rate of links are the ones that determine

pheromone values in terms of metrics

- Through this algorithm we try to find paths that are characterized by a degree of delay between S and D, and also the minimum number of hops and the maximum bandwidth links between the nodes.

- In complex networks it is extremely important (pheromone standards used to determine track quality)

• Using these composite pheromones allows multiple targets to be improved at the same time, which may be very important in complex networks (pheromone standards used to determine path quality)

Step 3: • Through each node in the network, agents are launched towards a destination node d in conjunction with data traffic at regular intervals.

These factors are known to be transmitted from their source to the front ants destination nodes.

• Each front ant is a randomized experiment designed to gather non-local information about the routes and traffic patterns of the contract.

Step 4: • The specific task for each front ant is to find the minimum delay path and the average maximum bandwidth link between the source and destination nodes.

Step 5: • The front ant collects information about travel time on the go. Once they reach the destination, the front ant becomes an ant back and returns to the source node via moving on the similar path.

Step 6: • The background ant updates the local routing information for each node in the path followed by the front ant to d in each node visited, and is associated with selecting the next step to access each node.

Step 7: The reaction is carried forward by an ant. When the source node initiates a session with the destination node d and no relevant information on how to access d is available, the node manager must collect long-term knowledge about the various pathways.

Step 8:

- Each ant keeps a running tally of the nodes it has visited.
- When destination d is reached, it is transformed into a backward ant, which returns the path to the source.

• Ants information is used to update the entry in the VERMON table, in each intermediate node, coming from a neighbor.

• Fermon variables between the nodes used, based on the adoption of the input update method on the minimum delay and the average bandwidth links.

Step 9:

• Managers in source nodes send proactive antipersonnel on a regular basis to update information about presently used paths and try to find new and maybe superior ones.

• During the communication session, they follow pheromones and upgrade pheromone tables in the same way as interactive ants do.

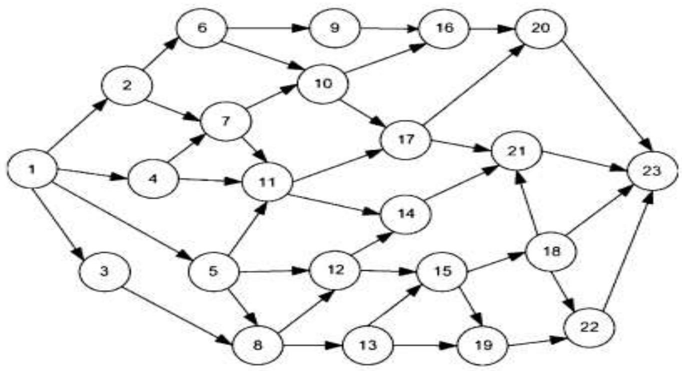
Step 10: • With proactive path improvement procedures, the router setup stage creates a network of multiple paths between source and destination.

The average minimum bandwidth connection of the delay path depends on the data rerouting as stated by the Stochastic plan and based on the values of the pheromone

• In each intermediate node, the random decision policy is applied to determine the next node to move to. The front ant moves from node to adjacent node towards its destination.

Example 1. The Direct graph.

To show the effectiveness of the existing model on type-2 Trapezoidal Fuzzy Numbers, we consider an example with twenty three vertices and each of which are connected at least to one of the vertexes on the given graph below.



[Figure 3: Twenty three nodes direct graph network]

Arc	Fuzzy weight	Minimum Delay	Maximum Band Width	Arc	Fuzzy weight	Minimum Delay	Maximum Band Width
d12 =A1(1-2)	(12,13,15,17)	14.166	12	d1017=A21(10-17)	(15,19,20,21)	19	19
d13 = A2(1-3)	(9,11,13,15)	12	11	d1114=A22(11-14)	(8,9,11,13)	10.166	10
d14 = A3(1-4)	(8,10,12,13)	10.833	9	d1117=A23(11-17)	(6,9,11,13)	9.833	9
d15 =A4(1-5)	(7,8,9,10)	8.5	8	d1214=A24(12-14)	(13,14,16,18)	15.16	16
d26 = A5(2-6)	(5,10,15,16)	11.833	11	d1215=A25(12-15)	(12,14,15,16)	14.33	23
d27 = A6(2-7)	(6,11,11,13)	10.5	9	d1315=A26(13-15)	(10,12,14,15)	12.83	12
d38 =A7(3-8)	(10,11,16,17)	13.5	13	d1319=A27(13-19)	(17,18,19,20)	18.5	18
d47 = A8(4-7)	(17,20,22,24)	20.833	20	d1421=A28(14-21)	(11,12,13,14)	12.5	15
d411 = A9(4-11)	(6,10,13,14)	44.33	11	d1518=A29(15-18)	(8,9,11,13)	10.16	8.5
d58= A10(5-8)	(6,9,11,13)	43.16	9	d1519=A30(15-19)	(5,7,10,12)	8.5	12
d511 = A11(5-11)	(7,10,13,14)	11.16	11	d1620=A31(16-20)	(9,12,14,16)	12.83	12
d512 = A12(5-12)	(10,13,15,17)	13.83	13	d1720=A32(17-20)	(7,10,11,12)	10.166	9
d69 = A13(6-9)	(6,8,10,11)	8.83	7	d1721=A33(17-21)	(6,7,8,10)	7.66	7
d610 = A14(6-10)	(10,11,14,15)	12.5	10	d1821=A34(18-21)	(15,17,18,19)	17.33	17
d710 = A15(7-10)	(9,10,12,13)	11	10	d1822=A35(18-22)	(3,5,7,9)	6	6
d711= A16(7-11)	(6,7,8,9)	7.5	7	d1823=A36(18-23)	(5,7,9,11)	8	8
d812 = A17(5-11)	(5,8,9,10)	8.166	8	d1922=A37(19-22)	(15,16,17,19)	16.66	16.66
d813 = A18(5-12)	(3,5,8,10)	6.5	6	d2023=A38(20-23)	(13,14,16,17)	15	15
D916 = A19(6-9)	(6,7,9,10)	8	8	d2123=A39(21-23)	(12,15,17,18)	15.66	15
d1016 = A20(10-16)	(12,13,16,17)	14.5	14	d2223=A40(22-23)	(4,5,6,8)	5.66	5

```
clc
% U of first example
U =[12 13 15 17; 9 11 13 15; 8 10 12 13; 7 8 9 10; 5 10 15 16; 6 11 11 13;
10 11 16 17; 17 20 22 24; 6 10 13 14; 6 9 11 13; 7 10 13 14; 10 13 15 17;
6 8 10 11; 10 11 14 15; 9 10 12 13; 6 7 8 9; 5 8 9 10; 3 5 8 10; 6 7 9 10;
12 13 16 17; 15 19 20 21; 8 9 11 13; 6 9 11 13; 13 14 16 18; 12 14 15 16;
10 12 14 15; 17 18 19 20; 11 12 13 14; 8 9 11 13; 5 7 10 12; 9 12 14 16;
7 10 11 12; 6 7 8 10; 15 17 18 19; 3 5 7 9; 5 7 9 11; 15 16 17 19;
13 14 16 17; 12 15 17 18; 4 5 6 8];

[n m]=size(U);
dmin=U(1,:);
for i=1:n
%-- a
% a=dmin(1,1)
% if U(i,1)< a
a=U(i,1);
b=U(i,2);
c=U(i,3);
d=U(i,4);
if U(i,4)==0
 SD(i)=1/6*(a+4*b+c)
else
SD(i)=1/6*(a+2*b+2*c+d)
end
end
```

```
SD =

  Columns 1 through 16

   14.1667   12.0000   10.8333    8.5000   11.8333   10.5000   13.5000   20.8333   11.0000    9.8333   11.1667   13.8333    8.8333   12.5000   11.0000    7.5000

  Columns 17 through 32

    8.1667    6.5000    8.0000   14.5000   19.0000   10.1667    9.8333   15.1667   14.3333   12.8333   18.5000   12.5000   10.1667    8.5000   12.8333   10.1667

  Columns 33 through 40

    7.6667   17.3333    6.0000    8.0000   16.6667   15.0000   15.6667    5.6667
```

```
% node 1 is the source and node 23 is the destination
% Minimum delay (msec)in direct graph.
G1=[0 14.16 12 10.83 8.5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;%1
0 0 0 0 0 11.83 10.5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;%2
0 0 0 0 0 0 0 13.5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;%3
0 0 0 0 0 0 20.83 0 0 0 11 0 0 0 0 0 0 0 0 0 0 0 0;%4
0 0 0 0 0 0 0 9.83 0 0 11.16 13.83 0 0 0 0 0 0 0 0 0 0 0;%5
0 0 0 0 0 0 0 0 8.83 12.5 0 0 0 0 0 0 0 0 0 0 0 0 0;%6
0 0 0 0 0 0 0 0 0 11 7.5 0 0 0 0 0 0 0 0 0 0 0 0;%7
0 0 0 0 0 0 0 0 0 0 0 8.166 6.5 0 0 0 0 0 0 0 0 0 0;%8
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 8 0 0 0 0 0 0 0;%9
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 14.5 19 0 0 0 0 0 0;%10
0 0 0 0 0 0 0 0 0 0 0 0 0 10.166 0 0 9.833 0 0 0 0 0 0;%11
0 0 0 0 0 0 0 0 0 0 0 0 0 15.16 14.33 0 0 0 0 0 0 0 0;%12
0 0 0 0 0 0 0 0 0 0 0 0 0 0 12.83 0 0 0 18.5 0 0 0 0;%13
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 12.5 0 0;%14
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 8.5 12.83 0 0 0;%15
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  0 0 12.83 0 0 0;%16
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 10.166 7.66 0 0;%17
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 17.33 6 8;%18
0 0 0 0 0 0 0 0 0 110 0 0 0 0 0 0 0 0 0 0 0 16.66 0;%19
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 15;%20
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 15.66;%21
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 5.66;%22
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0.5];%23
```

```
% node 1 is the source and node 23 is the destination
%Maximum Band width (kbps)in direct graph .
G2=[0 12 11 9 8 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;%1
0 0 0 0 0 11 9 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;%2
0 0 0 0 0 0 0 13 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;%3
0 0 0 0 0 0 20 0 0 0 11 0 0 0 0 0 0 0 0 0 0 0 0;%4
0 0 0 0 0 0 0 9 0 0 11 13 0 0 0 0 0 0 0 0 0 0 0;%5
0 0 0 0 0 0 0 0 7 10 0 0 0 0 0 0 0 0 0 0 0 0 0;%6
0 0 0 0 0 0 0 0 0 10 7 0 0 0 0 0 0 0 0 0 0 0 0;%7
0 0 0 0 0 0 0 0 0 0 0 8 6 0 0 0 0 0 0 0 0 0 0;%8
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 8 0 0 0 0 0 0 0;%9
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 14 19 0 0 0 0 0 0;%10
0 0 0 0 0 0 0 0 0 0 0 0 0 10 0 0 9 0 0 0 0 0 0;%11
0 0 0 0 0 0 0 0 0 0 0 0 0 16 33 0 0 0 0 0 0 0 0;%12
0 0 0 0 0 0 0 0 0 0 0 0 0 0 12 0 0 0 18 0 0 0 0;%13
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 15 0 0;%14
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 8.5 12 0 0 0;%15
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  0 0 12 0 0 0;%16
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 9 7 0 0;%17
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 17 6 8;%18
0 0 0 0 0 0 0 0 0 11 0 0 0 0 0 0 0 0 0 0 0 16.66 0;%19
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 15;%20
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 15;%21
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 5;%22
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0.5];%23
```

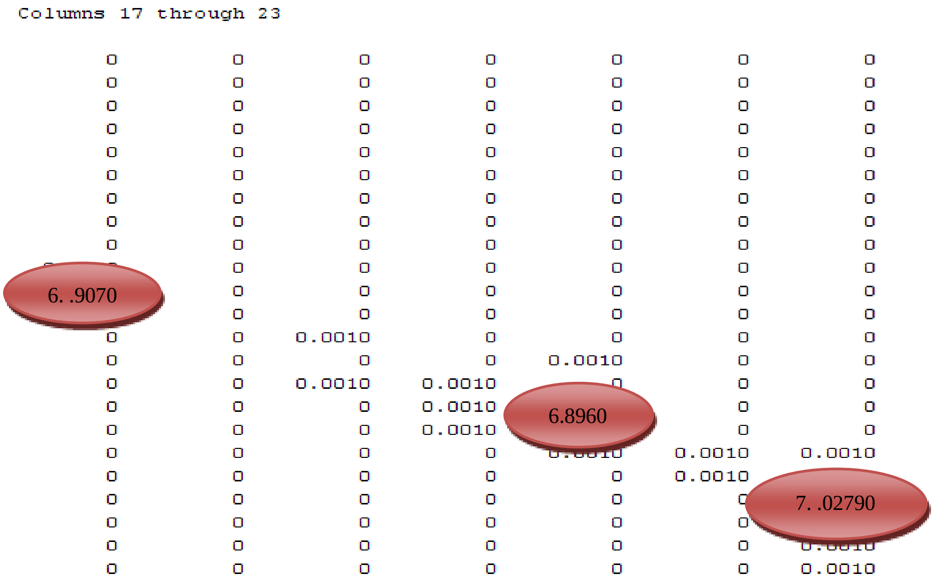
```
N=length(G); % the number of nodes in direct graph.
D=[0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;%1
0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;%2
0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;%3
0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0;%4
0 0 0 0 0 0 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0;%5
0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0;%6
0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0;%7
0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0;%8
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0;%9
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0;%10
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0;%11
0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0;%12
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 18 0 0 0 0;%13
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0;%14
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0;%15
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  0 0 1 0 0 0;%16
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0;%17
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1;%18
0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0;%19
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1;%20
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1;%21
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1%22
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1];%23
```

The output from matlab code

D =

Columns 1 through 16

0	0.0010	0.0010	0.001	6.334	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	[illegible]	.0010	0.0010	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0.0010	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0.0010	0	0	0	[illegible]	0	0	0	0	0
0	0	0	0	0	0	0	0.0010	0	[illegible]	6.769	010	0	0	0	0
0	0	0	0	0	0	0	0	0.0010	0.001	[illegible]	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0.0010	0.0010	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0.0010	0.0010	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0010
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.1086
0	0	0	0	0	0	0	0	0	0	0	0	0	0.0010	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0.0010	0.0010	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0010	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0.0010	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



Columns 17 through 23

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
[illegible]	0	0	0	0	0	0
[illegible]	0	0	0	0	0	0
[illegible]	0	0	0	0	0	0
0	0	0.0010	0	0	0	0
0	0	0	0	0.0010	0	0
0	0	0.0010	0.0010	0	0	0
0	0	0	0.0010	[illegible]	0	0
0	0	0	0.0010	[illegible]	0	0
0	0	0	0	0.0010	0.0010	0.0010
0	0	0	0	0	0.0010	[illegible]
0	0	0	0	0	0	[illegible]
0	0	0	0	0	0	[illegible]
0	0	0	0	0	0	0.0010
0	0	0	0	0	0	0.0010

In the outcome matrix of conductivity D, we find that elements $D_{1,5}$, $D_{5,11}$, $D_{11,17}$,$D_{17,21}$,$D_{21,23}$ reinforce the pheromone. So, the shortest path is 1 - 5 - 11 - 17 - 21 - 23. The total cost of the shortest path is 51.153, but by using Fuzzy Dijkstra algorithm and the elements bath $D_{1,5}$, $D_{5,12}$, $D_{12,15}$,$D_{15,18}$,$D_{18,22}$. So, the shortest path is 1 - 5 - 12 - 15 - 18 - 22. The total cost of the shortest path is 52.8333

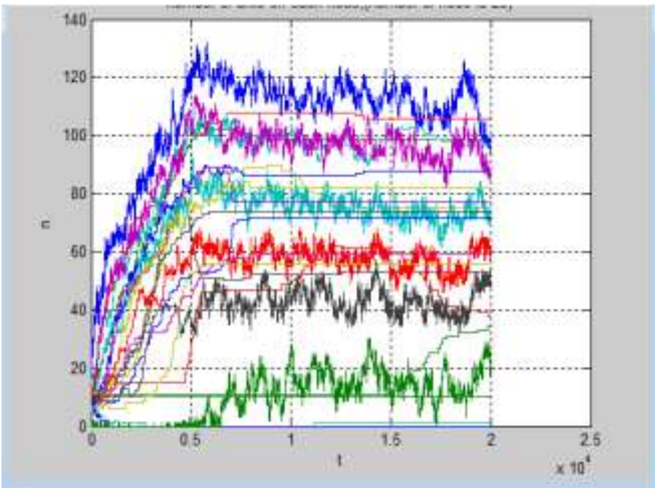


Figure 4: Number of ants on each node (number of nods 23)

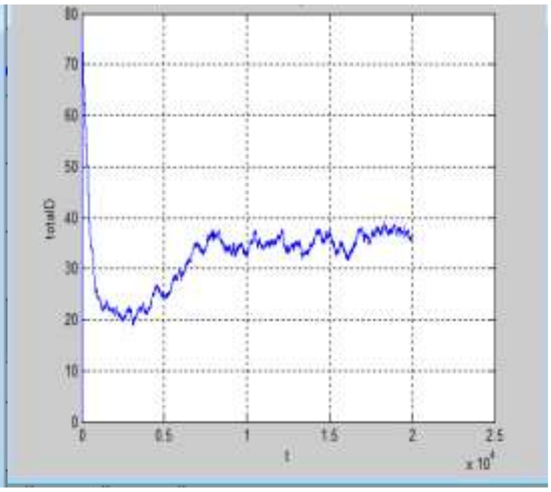


Figure 5: Total amount of pheromone

Example 2. The Indirect graph.

To show the effectiveness of the existing model on type-2 Trapezoidal Fuzzy Numbers, we consider an example with twenty three vertices and each of which are connected at least to one of the vertexes on the given graph below

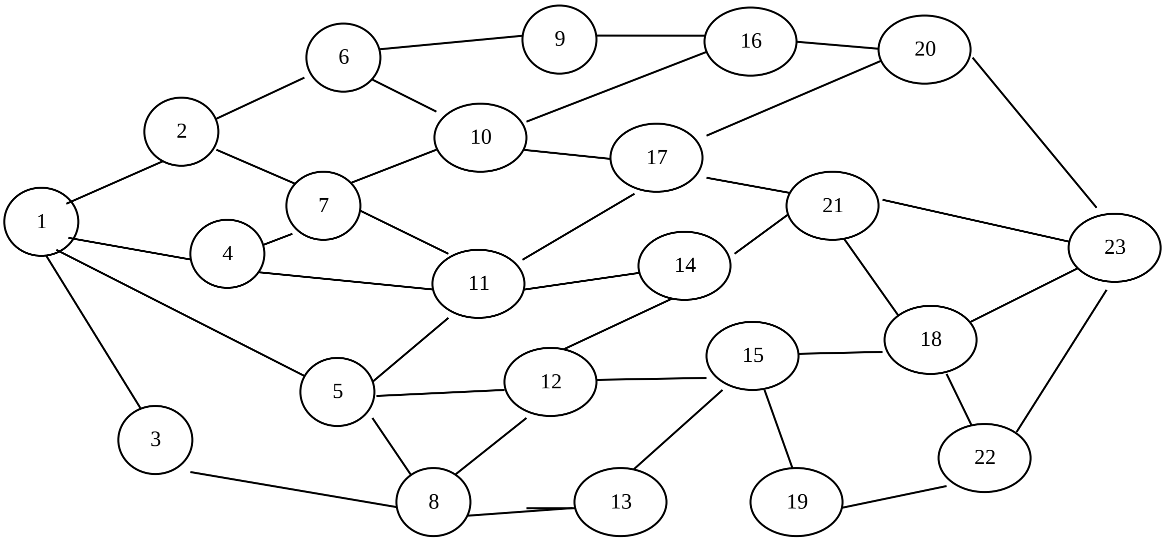


Figure 6: Twenty three nodes indirect graph network

```
% node 1 is the source and node 23 is the destination
% Minimum delay (msec)in indirect graph.
G1=[0 14.16 12 10.83 8.5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;%1
14.16 0 0 0 0 11.83 10.5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;%2
12 0 0 0 0 0 0 13.5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;%3
10.83 0 0 0 0 0 20.83 0 0 0 11 0 0 0 0 0 0 0 0 0 0 0 0;%4
8.5 0 0 0 0 0 0 9.83 0 0 11.16 13.83 0 0 0 0 0 0 0 0 0 0 0;%5
0 011.83 0 0 0 0 0 0 8.83 12.5 0 0 0 0 0 0 0 0 0 0 0 0 0;%6
0 10.5 0 20.83 0 0 0 0 0 11 7.5 0 0 0 0 0 0 0 0 0 0 0 0;%7
0 0 13.5 0 9.83 0 0 0 0 0 0 8.166 6.5 0 0 0 0 0 0 0 0 0 0;%8
0 0 0 0 0 8.33 0 0 0 0 0 0 0 0 0 8 0 0 0 0 0 0 0;%9
0 0 0 0 0 12.5 11 0 0 0 0 0 0 0 0 14.5 19 0 0 0 0 0 0;%10
0 0 0 11 11.16 0 7.5 0 0 0 0 0 0 10.166 0 0 9.833 0 0 0 0 0 0;%11
0 0 0 0 13.83 0 0 8.16 0 0 0 0 0 15.16 14.33 0 0 0 0 0 0 0 0;%12
0 0 0 0 0 0 0 6.5 0 0 0 0 0 0 12.83 0 0 0 18.5 0 0 0 0;%13
0 0 0 0 0 0 0 0 0 0 10.16 15.16 0 0 0 0 0 0 0 0 12.5 0 0;%14
0 0 0 0 0 0 0 0 0 0 0 14.33 12.83 0 0 0 0 10.16 8.5 0 0 0 0;%15
0 0 0 0 0 0 0 0 8 14.5 0 0 0 0 0 0 0  0 0 12.83 0 0 0;%16
0 0 0 0 0 0 0 0 0 19 9.83 0 0 0 0 0 0 0 0 10.166 7.66 0 0;%17
0 0 0 0 0 0 0 0 0 0 0 0 0 0 10.16 0 0 0 0 0 17.33 6 8;%18
0 0 0 0 0 0 0 0 0 0 0 18.5 0 0 8.5 0 0 0 0 0 0 16.66 0;%19
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 12.83 7.66 0 0 0 0 0 15;%20
0 0 0 0 0 0 0 0 0 0 0 0 0 12.5 0 0 7.66 0 0 0 0 0 15.66;%21
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 6 16.66 0 0 5.66 0;%22
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 8 0 15 15.66 5.66 0];%23
```

```
% node 1 is the source and node 23 is the destination
%Maximum Band width (kbps)in indirect graph .
G2=[0 12 11 9 8 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;%1
12 0 0 0 0 11 9 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;%2
11 0 0 0 0 0 0 13 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;%3
9 0 0 0 0 0 20 0 0 0 11 0 0 0 0 0 0 0 0 0 0 0 0;%4
8 0 0 0 0 0 0 9 0 0 11 13 0 0 0 0 0 0 0 0 0 0 0;%5
0 11 0 0 0 0 0 0 7 10 0 0 0 0 0 0 0 0 0 0 0 0 0;%6
0 9 0 20 0 0 0 0 0 10 7 0 0 0 0 0 0 0 0 0 0 0 0;%7
0 0 13 0 9 0 0 0 0 0 0 8 6 0 0 0 0 0 0 0 0 0 0;%8
0 0 0 0 0 7 0 0 0 0 0 0 0 0 0 8 0 0 0 0 0 0 0;%9
0 0 0 0 0 10 10 0 0 0 0 0 0 0 0 14 19 0 0 0 0 0 0;%10
0 0 0 11 11 0 7 0 0 0 0 0 0 10 0 0 9 0 0 0 0 0 0;%11
0 0 0 0 13 0 0 8 0 0 0 0 0 16 33 0 0 0 0 0 0 0 0;%12
0 0 0 0 0 0 0 6 0 0 0 0 0 0 12 0 0 0 18 0 0 0 0;%13
0 0 0 0 0 0 0 0 0 0 10 16 0 0 0 0 0 0 0 0 15 0 0;%14
0 0 0 0 0 0 0 0 0 0 0 16 33 0 0 0 0 8.5 12 0 0 0 0;%15
0 0 0 0 0 0 0 0 8 14 0 0 0 0 0 0 0  0 0 12 0 0 0;%16
0 0 0 0 0 0 0 0 0 19 9 0 0 0 0 0 0 0 0 9 7 0 0;%17
0 0 0 0 0 0 0 0 0 0 0 0 0 0 8.5 0 0 0 0 0 17 6 8;%18
0 0 0 0 0 0 0 0 0 0 0 0 18 0 12 0 0 0 0 0 0 16.66 0;%19
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 12 9 0 0 0 0 0 15;%20
0 0 0 0 0 0 0 0 0 0 0 0 0 15 0 0 7 0 0 0 0 0 15;%21
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 6 16.66 0 0 0 5;%22
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 8 0 15 15 5 0];%23
```

```
N=length(G); % the number of nodes
D=[0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;%1
1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;%2
1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;%3
1 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0;%4
1 0 0 0 0 0 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0;%5
0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0;%6
0 1 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0;%7
0 0 1 0 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0;%8
0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0;%9
0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0;%10
0 0 0 1 1 0 1 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0;%11
0 0 0 0 1 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0;%12
0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0;%13
0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 0;%14
0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1 1 0 0 0 0;%15
0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0  0 0 1 0 0 0;%16
0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 1 0 0;%17
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1 1;%18
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0;%19
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1;%20
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 1;%21
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 1;%22
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 0];%23
```

The output from matlab code

Columns 1 through 16

0	0.0010	0.0010	0.00[illegible]	7.6758	0	0	0	0	0	0	0	0	0	0	0
0.0010	0	0	0	0	0.0010	0.0010	0	0	0	0	0	0	0	0	0
0.0010	0	0	0	0	0	0	0.0010	0	0	0	0	0	0	0	0
0.0010	0	0	0	0	0	0.0010	0	0	[illegible]	7.1992	0	0	0	0	0
0.0010	0	0	0	0	0	0	0.0564	0	[illegible]	[illegible]	[illegible]	0	0	0	0
0	0.0010	0	0	0	0	0	0	0.0010	0.0010	0	0	0	0	0	0
0	0.0010	0	0.0010	0	0	0	0	0	0.0010	0.0010	0	0	0	0	0
0	0	0.0010	0	0.0010	0	0	0	0	0	0	0.0010	0.0211	0	0	0
0	0	0	0	0	0.0010	0	0	0	0	0	0	0	0	0	0.0010
0	0	0	0	0	0.0010	0.0010	0	0	0	0	0	0	0	0	0.0010
0	0	0	0.0010	0.0010	0	0.0010	0	0	0	0	0	0	0.0010	0	0
0	0	0	0	0.0010	0	0	0.0010	0	0	0	0	0	0.0010	0.0010	0
0	0	0	0	0	0	0	0.0010	0	0	0	0	0	0	0.0179	0
0	0	0	0	0	0	0	0	0	0	0.0010	0.0010	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0.0010	0.0010	0	0	0
0	0	0	0	0	0	0	0	0.0010	0.0010	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0.0010	0.0010	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0010	0
0	0	0	0	0	0	0	0	0	0.0010	0	0.0010	0	0	0.0010	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0010
0	0	0	0	0	0	0	0	0	0	0	0	0	0.0010	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Columns 17 through 23

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0.0010	0	0	0	0	0	0
7.2955	0	0	0	0	0	0
[illegible]	0	0	0	0	0	0
[illegible]	0	0.0010	0	0	0	0
0	0	0	0	0.0010	0	0
0	0.1215	0.0010	0.0010	0	0	0
0	0	0	0.0010	0	0	0
0	0	0	0.0010	6.8443	0	0
0	0	0	0	[illegible]	0.0010	0.0215
0	0	0	0	[illegible]	0.0010	0
0.0010	0	0	0	[illegible]	0	0.0010
0.0010	0	0	0	0	0	6.2748
0	0.0010	0.0010	0	0	0.0010	[illegible]
0	0.0010	0	0.0010	0.0010	0.0010	[illegible]

7.2955

6.8443

6.2748

In the outcome matrix of conductivity D, we find that elements $D_{1,5}$, $D_{5,11}$, $D_{11,17}$,$D_{17,21}$,$D_{21,23}$ reinforce the pheromone. So, the shortest path is 1 - 5 - 11 - 17 - 21 - 23.

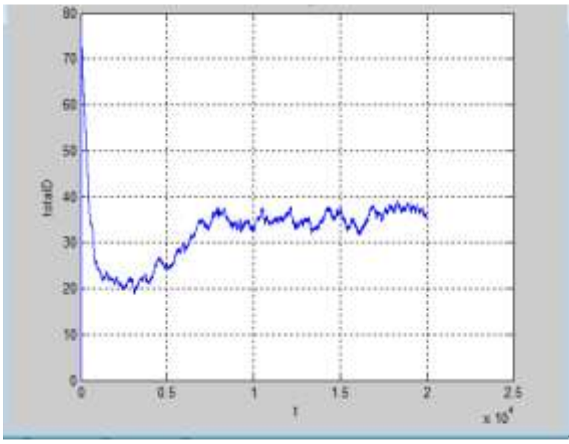


Figure 7: Number of ants on each node (number of nods 23)

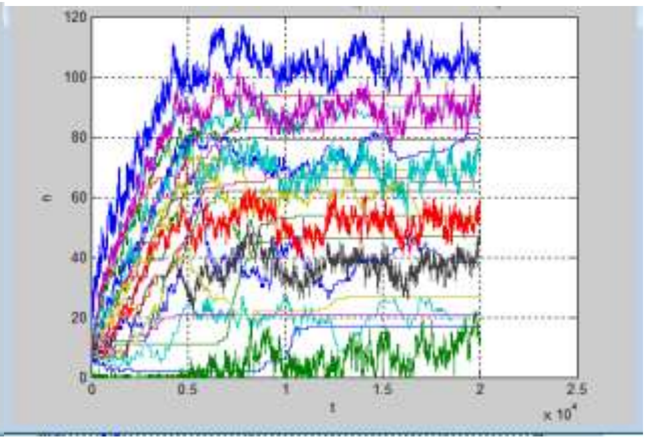


Figure 8: Total amount of pheromone

5. CONCLUSION

The problem of finding a shorter path in the network has been resolved in many ways by our goals. In this summary, we presented two different ways to solve this problem. The first is using Ant Colony Optimization (ACO) where the routing protocol was searched with network encryption and the proposed network performance was displayed in terms of packet delay, throughput consumption, and bandwidth in the direct network. The second study Ant Colony Optimization (ACO) where the routing protocol was searched with network encryption and the proposed network performance was displayed in terms of packet delay, throughput consumption, and bandwidth in the indirect network. Also, Compromise this result by the result of the Fuzzy Dijkstra algorithm[10]. We conclude from this study that the new method gives the same results in the case of direct and undirect networks and also gives better results of the Fuzzy Dijkstra algorithm for shortest path problem under uncertain environment [12].

REFERENCES

[1] Yang , X. , Liu, B. , *A Journal of Modeling and Computation Under Uncertainty* , 18 , 263–278 , 2019 , DOI: 10.4018/978-1-5225-5709-8.ch014

[2] Ardekani , M. R. H. , Abbasi , M. , *Computational Intelligence for Modelling and Prediction* , 10 (1) , 286-294 , 2015 , DOI: 10.1007/10966518_7

[3] Gen, M. , Lin , L. , *GECCO* , 06. Pp. 1411-1412 , 2006, DOI: 10.1145/1143997.1144220

[4] Geetha, R. , Srikanth , G. , *International Research Journal of Mobile and Wireless Communications*, 03(01) , 2012, Downloaded from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.258.9921&rep=rep1&type=pdf

[5] Sabri, D., *Brunel University Research Archive(BURA)* , 2011 , Downloaded from https://bura.brunel.ac.uk/handle/2438/6435

[6] Dorigo, M., Maniezzo, V. , Colorni , A. , *IEEE Transactions on Systems*, Part B, 26(1) : 29-41, 1996 , DOI: 10.1109/3477.484436

[7] Dorigo,M. , Bonabeau, E., Theraulaz, G. , *Future Gener Comput. Syst*., 16(8) 851-871 , 2000, DOI: 10.1016/S0167-739X(00)00042-X

[8] Hu , X. , Zhang, J. , *Mathematical Problems in Engineering*, Article ID 432686 , 2013, DOI: 10.1155/2013/432686

[9] Di Caro , G. A., Ducatelle , F. , Gambardella , L. M. , *The Complex Coevolution of Information Technology Ecosystems*, 32 (2008) , DOI: 10.4018/978-1-59904-627-3

[10] Denga ,Y., Chenb,Y. , Zhanga, Y., Mahadevanc , S. , *Applied Soft Computing* , 1231–1237 , 2012 , DOI: 10.1016/j.asoc.2011.11.011

[11] Yousif, E. , Salama, A. , Wahed, M .E. , *International Conference on Science and Technology* , 2019 Downloaded from http://confs.naturalspublishing.com/icst2018/Abstracts.asp

[12] Salama , A.A., Wahed , M . E., Yousif , E., *Neutrosophic Knowledge - University of New Mexico*, 1, 41-53, 2020 , DOI: 10.5281/zenodo.4269558

[13] Zakzouk , A. , Zaher , H. , Zean El-Deen , R., *IEEE Xplore* , 2010 , DOI : 10.1109/ICNC.2012.6234645

[14] Gasbaoui, B., Chaker, A., Laoufi, A., Abderrahman, A., Allaoua, B., *Leonardo Electronic Journal of Practices and Technologie*s , 9(16), 75–88 , 2010 , Downloaded from https://link.springer.com/chapter/10.1007/978-3-642-32063-7_23

[15] Baterina, A.V., Oppus, C., *International Journal of Circuits* , 4(2), 25–33 , 2010 , Downloaded from https://archium.ateneo.edu/discs-faculty-pubs/202/

[16] Rezapour, O.M., Dehghani, A., Shui, L.T. , *Australian Journal of Basic and Applied Sciences* ,4 (7) , 2099-2108 , 2010 , Downloaded from https://www.cabdirect.org/cabdirect/abstract/20103359587

[17] Bella, J. , Mullenb , M., *Advanced Engineering Informatics* , 18, 41–48 , 2004 , DOI: 10.1016/j.aei.2004.07.001

[18] Salehinejad, H. , Talebi, *Applied Computational Intelligence and Soft Computing* , 1(13) , 2010 , DOI: 10.1155/2010/428270

[19] Toksari, *Applied Mathematics and Computation* 176, 308–316 , 2006 , DOI : 10.1016/j.amc.2005.09.043

[20] Dorigo, M. , Blumb, *Theoretical Computer Science* , 344, 243–278 , 2005 , DOI : 10.4249/scholarpedia.1461

[21] Dorigo, M. , Maniezzo, V. , Colorni, *IEEE Transactions on Systems* , Part B 26(1), 1–13 , 1996 , Downloaded from https://ieeexplore.ieee.org/abstract/document/484436

[22] Georgilakis, Vernados, Karytsas, *Journal of Optoelectronics and Advanced Materials* , 10(5), 1246–1250 , 2008 ,DOI: 10.1016/j.eswa.2011.02.027

[23] Bouhafs, L. , Hajjam, A., Koukam, *Journal of Artificial Intelligence - Theory and Application* ,1(1), 31–34 , 2010 , DOI: 10.1109/3477.484436

[24] Saeheaw, T. , Charoenchai, N. , Chattinnawat, *World Academy of Science, Engineering and Technology* , 60, 654–659 , 2009 , DOI: 10.1016/j.cie.2007.08.003

[25] Thangavel, Karnan, Jeganathan, lakshmi, P. , Sivakumar, Geetharamani , *ACSE Journal* 6(1), 7–26 , 2006 , DOI : 10.1109/ICNC.2012.6234645

[26] Tripathi, M. , Kuriger, G., Wan, H.D. , *Proceedings of the Winter Simulation Conference* , 2476–2487 , 2009 , DOI: 10.1109/WSC.2009.5429417

[27] Maria, L., Stanislav, *Journal of Applied Mathematics* , 2(2), 229–238 , 2009 , DOI: 10.1109/3477.484436

[28] [28] Chowdhury ,S. , Marufuzzaman , M. , Tunc , H. , Bian , Bullington , L. , W. , *Journal of Computational Design and Engineering* , 6(3) , 2019 , DOI : 10.1016/j.jcde.2018.10.004