Solving fuzzy shortest path problem by genetic algorithm

A Syarif[1], K Muludi[1], R Adrian[1], M Gen[2]

[1]Department of Computer Science, Faculty of Mathematics and Natural Sciences
Lampung University, Indonesia, Jl. Brodjonegoro No. 1, Bandar Lampung,35145
[2]Fuzzy Logic Institute, Tokyo, Japan.

Corresponding author: admi_syarif@yahoo.com

Abstract: Shortest Path Problem (SPP) is known as one of well-studied fields in the area Operations Research and Mathematical Optimization. It has been applied for many engineering and management designs. The objective is usually to determine path(s) in the network with minimum total cost or traveling time. In the past, the cost value for each arc was usually assigned or estimated as a deteministic value. For some specific real world applications, however, it is often difficult to determine the cost value properly. One way of handling such uncertainty in decision making is by introducing fuzzy approach. With this situation, it will become difficult to solve the problem optimally. This paper presents the investigations on the application of Genetic Algorithm (GA) to a new SPP model in which the cost values are represented as Triangular Fuzzy Number (TFN). We adopts the concept of ranking fuzzy numbers to determine how good the solutions. Here, by giving his/her degree value, the decision maker can determine the range of objective value. This would be very valuable for decision support system in the real world applications.Simulation experiments were carried out by modifying several test problems with 10-25 nodes. It is noted that the proposed approach is capable attaining a good solution with different degree of optimism for the tested problems.

Keywords: Shortest Path Problem, Network Design Problem, Genetic Algorithm, Ranking Fuzzy Number, Decision Support System

1. Introduction

In business management, many important problems are expressed as network design problems. Among all classes of problems in network optimization, Shortest Path Problem (SPP) has been one of the most extensively disscused in computer science literature. It is commonly encountered in wide array of practical applications including transportation planning, communication, and production applications, salesperson routing and often imbedded within other types of network optimization problems [1], [2].

Given a weighed network graph, the SPP is to obtain the minimum total weight path(s) between nodes in the network. In the real world applications, the weight value represents the average delivery time of the commodities, reliability of transportation, and accessibility to the users, product deterioration, and so on. Thus, finding a good solution of SPP would be very important and could save millions of transportation and/or industrial cost, yet computationally difficult task. There have been several methods for solving traditional SPP including Dijkstra method [3]. However, due to the intrinsic difficulty, many authors introduced search methods based upon heuristics to give solutions within reasonable computational time [4], [5].

Traditionally, SPP was investigated in the area of Operations Research and Artificial Intelligence that is modelled by certain/deterministic cost values. However, for many real world applications, the deterministic model can be far from sufficient. For instance, in a transportation network, the path that has the least travel time, during the rush hours would have traffic jams that may lead to a drastic increase in travel time. One way of handling such uncertainty in decision making is by introducing fuzzy programming approach [6].

Artificial Intelligence (AI) has been discussed in many recent research fields. It has been successfully used to solve many difficult-real-world optimization problems. Among them, Genetic algorithm (GA) has been known as one of effective and popular methods. Since introduced by Holland [7], it has been implemented to give optimal or near optimal solution for many NP-hard optimization problems with less computational cost [8]. There are several advantages of using GA as follows: It deals with coding instead of decision variables. It requires no domain knowledge - only the payoff or objectives. In addition, traditional methods use deterministic transition rules to guide the search, such as hill-climbing, and neighbourhood search. Another difference between traditional methods and genetic algorithms is: the latter searches from a set of points, while the former from a single point. This makes GA more robust than traditional methods regarding their potential as optimization techniques [9]. In our previous works, we also have successfully implemented GA to solve various combinatorial optimization problems and found that GA can give very good solutions to the problems within reasonable computational time [10], [11], [12], [13].

In this paper, we develop new model of SPP called fuzzy Shortest Path Problem (f-SPP). Here, the cost values of the f-SPP are represented by using Triangular Fuzzy numbers (TFN). Our major efforts in this work include the adaption of the the rangking method for handing the fuzziness. We adopt the concept of the ranking Fuzzy numbers with integral value for the evaluation and selection. In this technique, the decision maker can determine his decision by giving flexible value for the degree of optimism [14].

The remainder of this paper is organized as follows: The description of the mathematical formulation of the f-SPP is given in Section 2. Section 3 describes a brief discussion design of algorithm. It includes ranking Fuzzy numbers, representation of chromosome, genetic operations and selection method. Section 4 presents the numerical experiments and results obtained by the proposed method. Finally, some concluding remarks are given in Section 5.

2. Mathematical Model

In this section, we shall present a comprehensive mathematical model of the problem. Suppose that we are given a graph G in which thr lenght of each arc has associated with a TFN. In some applications, that TFN may actually represent traveling time or some other values. For any two vertices, s and t in G, it is possible that there exist several paths from s to t. The problem involves finding path(s) from s to t that has the shortest traveling time.

The model of f-SPP can mathematically be stated as follows:

$$\textbf{min} \quad \sum_{i=1}^{m}\sum_{j=1}^{n}\tilde{c}_{ij}x_{ij} \quad \cdots\cdots \quad (1)$$

$$\textbf{s.t.} \quad \sum_{j=1}^{n}x_{ij}-\sum_{k=1}^{n}x_{ki}$$

$$=\begin{cases} 1 \quad (i=1) \quad \cdots\cdots & (2)\\ 0 \quad (i=2,3,\cdots,n-1) \quad \cdots\cdots & (3)\\ -1 \quad (i=n) \quad \cdots\cdots & (4)\end{cases}$$

$$x_{ij} = \begin{cases} 1, & \text{if link } (i, j) \text{ is included in the path} \\ 0, & \text{otherwise} \end{cases}$$

In the above model, directed graph ***G***=(***V***, ***A***) where ***V*** is a set of nodes, ***A*** is a set of links. $\tilde{c}_{ij}$ is a TFN representing fuzzy cost associated with each arc(*i*, *j*). Source node and destination node are node 1 and node *n* respectively. The length of a path is defined as the sum of the lengths of all individual arcs comprising the path.

3. Design of the Proposed Algorithm

3.1. Ranking Fuzzy Number (TFN)

When considering an optimization problem that its coefficients are represented with TFNs, the objective values of the problem will also be TFNs. In a fuzzy environment, an effective, efficient, and accurate ranking method becomes a very important decision making procedure. Many authors have investigated various ranking methods for fuzzy numbers [15]. The ranking fuzzy number with integral value technique was proposed by Liou & Wang [14]. The basic concept of this technique is to rank the TFN based on its total integral value that represents a ”mean value”. The left and right integral values are used to reflect the pessimistic and optimistic viewpoint of decision maker respectively. The total integral value is then computed based on this degree of optimism and each objective function values. A parameter α is given to adjust the degree of optimism.

Definition 1 [14]
Let $\tilde{A}$ be a TFN with its membership function $\mu_{\tilde{A}}$ as shown in Figure 1. A TFN is denoted by a triplet (a_1, a_2, a_3) where a_1, a_2, a_3 are real numbers. i.e. a minimum value, a modal value and a maximum value. The membership function of a TFN $\tilde{A}$, $\mu_{\tilde{A}}$ is defined as:

$$\mu_{\tilde{A}}(x) = \begin{cases} 0 & , x < a_1 \\ \dfrac{(x - a_1)}{(a_2 - a_1)} & , a_1 \le x \le a_2 \\ \dfrac{(a_3 - x)}{(a_3 - a_2)} & , a_2 \le x \le a_3 \\ 0 & , x > a_3 \end{cases} \quad \text{..} \quad (5)$$

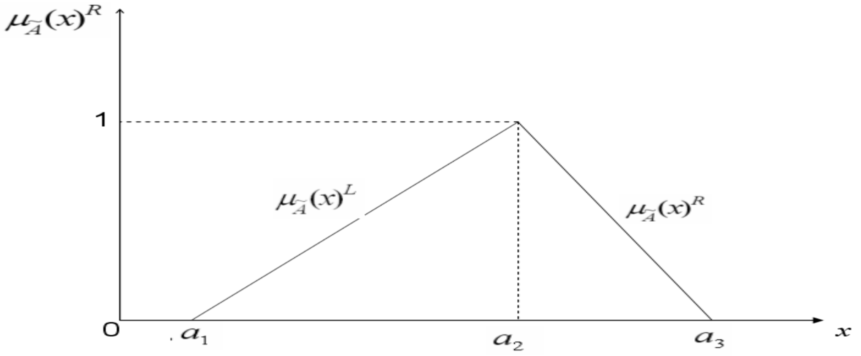


Figure 1. The membership function of TFN

From the definition of the above membership function, we can see that the left membership function $\mu_{\tilde{A}}(x)^L$ is continuous and strictly increasing. Its inverse function $g_{\tilde{A}}(x)^L$ would exist and continuous on interval [0, 1]. So we can compute its integral value on that interval. Similarly for the right membership function $\mu_{\tilde{A}}(x)^R$. We can compute both left and right integral values as follows:

$$I(\tilde{A})^L = \int_0^1 g_{\tilde{A}}(x)^L dy = \frac{1}{2}(a_1 + a_2) \quad \cdots\cdots \quad (6)$$

.

$$I(\tilde{A})^R = \int_0^1 g_{\tilde{A}}(x)^R dy = \frac{1}{2}(a_2 + a_3) \quad \cdots\cdots \quad (7)$$

Thus, the total integral value for triangular fuzzy number $\tilde{A}$ is

$$\begin{aligned} I_T^\alpha(\tilde{A}) &= \alpha I(\tilde{A})^R + (1-\alpha) I(\tilde{A})^L \quad \cdots\cdots \\ &= \frac{1}{2}[\alpha\, a_3 + a_2 + (1-\alpha)\, a_1] \quad \cdots\cdots \end{aligned} \quad (8)$$

The following definition is used in relation to the ranking of the TFN:

Definition 2: Suppose that $S = \{\tilde{A}_1, \tilde{A}_2, \cdots, \tilde{A}_n\}$ is a set of convex fuzzy numbers and that ranking fuzzy number *R* is a mapping from *S* to the real numbers. For any distinct $\tilde{A}_i, \tilde{A}_j \in S$ the ranking has the following properties.

1. if $R(\tilde{A}_i) < R(\tilde{A}_j)$ then $\tilde{A}_i < \tilde{A}_j$
2. if $R(\tilde{A}_i) = R(\tilde{A}_j)$ then $\tilde{A}_i = \tilde{A}_j$
3. if $R(\tilde{A}_i) > R(\tilde{A}_j)$ then $\tilde{A}_i > \tilde{A}_j$

Based on the above definition, thus for any fuzzy numbers $\tilde{A}_i$ and $\tilde{A}_j$ we shall have:

if $I_T^\alpha(\tilde{A}_i) < I_T^\alpha(\tilde{A}_j)$ then $\tilde{A}_i < \tilde{A}_j$
if $I_T^\alpha(\tilde{A}_i) = I_T^\alpha(\tilde{A}_j)$ then $\tilde{A}_i = \tilde{A}_j$
if $I_T^\alpha(\tilde{A}_i) > I_T^\alpha(\tilde{A}_j)$ then $\tilde{A}_i > \tilde{A}_j$.

3.2. Design of Genetic Algorithm (GA)

Generally, the behaviour of GAs depends on many uncertain factors. One of the important factors is the balance between exploitation and exploration in the search space. The balance is strongly affected by design strategy for GA and GA parameter values. Here, we would like to describe the consideration underlying the design of GA approach including representation method, genetic operators and selection method.

3.2.1. Chromosome Representation and Evaluation

It is well known that the first step in applying GA is to find a representation for a possible solution. An efficient design of the chromosome would lead to a good performance of GA.

For this problem, we adopt permutation-based representation. In this representation, the value of gene represents the priority of the node number for being included in the arc. There are several advantages of using this representation for network problems [9]. Those include, first, any permutation of the encoding corresponds to a path (legality). Most existing genetic operators can be easily applied to the encoding. Any path has a corresponding encoding (completeness). And, any point in solution space is accessible for genetic search.

The following Figure 2 shows the illustration of priority based representation.

7	2	5	1	4	3	9	8	6

Figure 2. Priority-based chromosome

In order to compute the fitness value, we decode the above chromosome using the decoding procedure as follows:

```
Procedure: Decoding
Step 1:
    - number of nodes $n$,
    - the alpha value
    - the set of nodes $S_i$ with all nodes adjacent to node $i$.
    - the set of the TFN weight values denoted by $(a_1, a_2, a_3)$
Step 2
    Determine path corresponding to the chromosome as follows:
    initial source node $i \leftarrow 1$, $P_k \leftarrow \phi$;
    while $S_i \neq \phi$ do
        select $l$ from $S_i$ with the highest priority;
        if $v_k(l) \neq 0$ then $v_k(l)=0$;
          $P_k \leftarrow P_k \cup \{x_{il}\}$;
          $i \leftarrow l$;
        else $S_i \leftarrow S_i \backslash \{l\}$
    end
    output the complete path $P_k$ ;
end
Step 3
    Compute the fuzzy objective values
Step 4
    Determine the integral values of TFNs
```

3.2.2. Crossover and Mutation

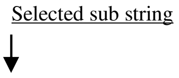
The next step of GA is to select chromosome for replication by using genetic operator, in order to have new chromosome to be evaluated further. Basically, GA use two genetic operators, crossover and mutation.

Crossover is known as the most important recombination operator in GA. There have been many variants of crossover operations given in the literature for permutation based representation such as: PMX method (Partially Matched Crossover) [16], PX (Position-based crossover) [17] and WMX (Weight mapping crossover) [18].

In this paper, we adopt PMX crossover operations as follows:

Procedure : PMX

Step 1: Select a section of chromosome randomly



Parent 1	1	7	2	3	4	6	5	8
Parent 2	4	6	3	5	7	1	8	2

Step 2: Exchanged each sub string

Parent 1	1	7	3	5	7	6	5	8
Parent 2	4	6	2	3	4	1	8	2

Step 3: Determine the mapping of genes in each substring

3	5	7
2	3	4

⇨

2 ⇔ 3 ⇔ 5

4 ⇔ 7

Step 4: Update chromosome with information given by Step 3

Offspring 1	1	4	3	5	7	6	2	8
Offspring 2	7	6	2	3	4	1	8	5

Mutation is usually done to prevent premature loss of information by exchanging the information within a chromosome. In this paper, we adopt the inversion mutation. It is done by selecting two positions within a chromosome at random and then inverts the sub-string between these two positions. We show the illustration of this mutation operation in Figure 3

Parent	4	6	3	5	7	1	8	2
offspring	4	6	7	5	3	1	8	2

Figure 3. Illustration of inversion mutation

3.2.3. Selection

When implementing GA, the way to select chromosome for the next generation is also very important issue. It should provide a balance between exploitation and exploration to search solution in search space. Here, we adopt one of the fitness-proportional methods called *roulette wheel* selection [2]. If the best chromosome is not selected for the next generation, we select one of the chromosome randomly and replace it with the best chromosome. We express our selection method as follows:

```
Procedure: Selection
Step 1: Calculate a cumulative probability $a_p$ for each chromosome
        $X_p, (p = 1, 2, \cdots, pop\_size)$.
Step 2: Generate a random real number $r$ in [0,1].
Step 3: If $r \le a_1$, then select the first chromosome $X_1$; otherwise select the $p^{th}$ chromosome
        $X_p$, $(2 \le p \le pop\_size)$ such that $a_{p-1} < r < a_p$.
Step 4: Repeat Steps 2 and 3 pop_size times and obtain pop_size copies of chromosome.
Step 5: If the best chromosome is not selected in the next generation, replace one randomly
        from the new population by the best one.
```

3.2.4. Overall Procedure

The overall GA procedure used in this paper is illustrated as in Figure 4.

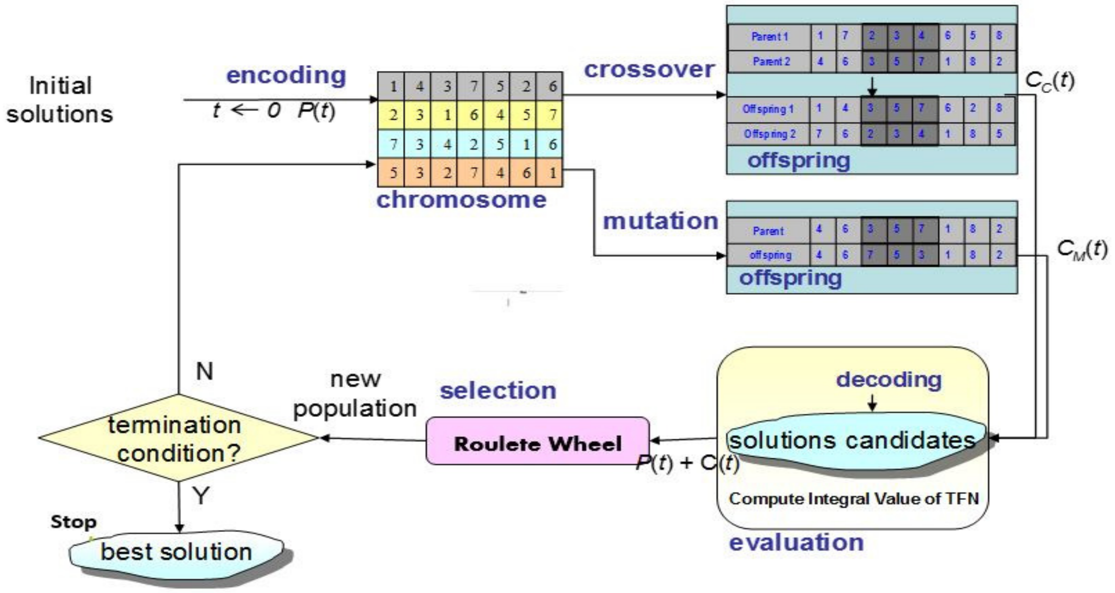


Figure 4. Overall procedure of proposed method

4. Experimental Design and Results

To explore how well the proposed GA can solve f-SPP. we have designed several numerical experiments. The proposed method was implemented by Using C language and run on PC Core i3. For the test problems, we modified the SPP test problems given [19], [2] and [20]. The size of test problems and GA parameters used for these experiments are set as presented in Table 1:

Table 1. Problem size and GA parameter values

Test Problem	Number of Node	Crossover Probability	Mutation Probability	Population Size	Maximimum generation
1	20	0.4	0.2	20	100
2	10	0.4	0.2	10	100
3	25	0.4	0.2	15	100

The first experiments were cunducted to see the effectiveness of GA with respect to the heterogeneity of the degree of optimism. To investigate this issue, for all test problems, we take a look at the variation of the degree of optimisms (0, 0.1, 0.2, …, 1). For each different degree of optimism, the experiment was done 10 (ten) times. We note the best, the average and the worst results. The experimental results are summarized in Table 2.

Table 2. Computational results for different alpha values

	Alpha										
Problem 1	**0.0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**
Maximum	152,5	164,6	176,7	188,8	200,9	213	232,6	237,2	249,3	261,4	273,5
Minimum	112,5	122,1	131,7	141,3	150,9	160,5	170,1	179,7	189,3	198,9	208,5
Average	120,5	130,6	140,7	155,55	160,9	171	187,35	191,2	201,3	205,2	221,5
STDEV	16,86	17,91	18,97	22,94	21,08	22,13	27,85	24,24	25,29	19,76	27,41
Problem 2	**0.0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**
Maximum	94	95,4	96,8	98,2	99,6	101	102,4	103,9	107,3	107,3	109
Minimum	92	93,7	95,4	97,1	98,8	100,5	102,2	103,8	105,2	106,6	108
Average	92,6	94,21	95,96	97,21	99,04	100,65	102,24	103,8	105,44	106,7	108,3
STDEV	0,96	0,82	0,72	0,34	0,38	0,24	0,08	0,042	0,66	0,29	0,483
Problem 3	**0.0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**
Maximum	89	88,35	90,7	92,8	95,4	98	100,6	103,2	104,8	107,2	110
Minimum	85	87,6	90	92,5	94,5	96,75	99	101,3	103,5	105,8	108
Average	85,85	87,825	90,07	92,56	94,59	97	99,32	101,4	103,63	106,2	108,2
STDEV	1,31	0,36	0,22	0,12	0,28	0,52	0,67	0,617	0,41	0,67	0,632

To observe the speed of convergence of the proposed GA, we noted the objective value in each generation for the test problem 3. We compared the results for three different alpha values (0, 0.5 and 1). The results are shown in Figure 5.

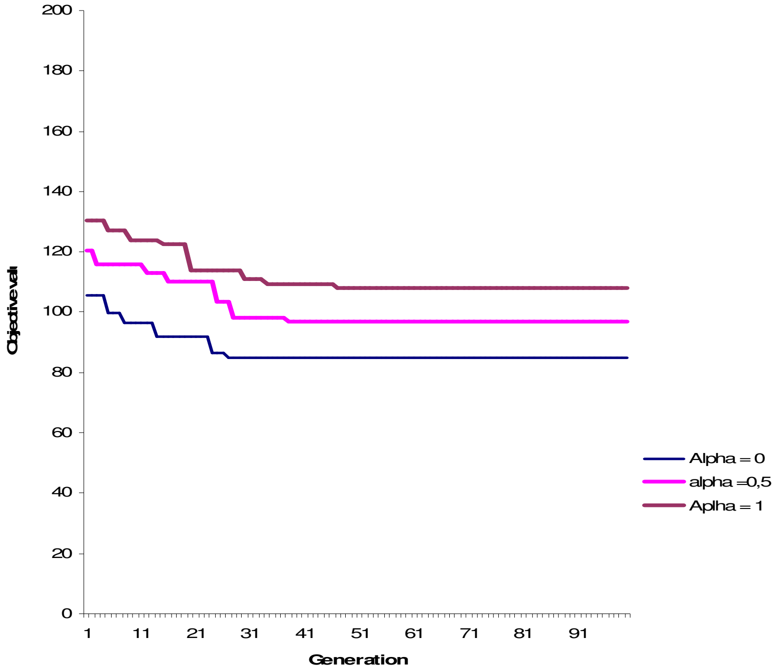


Figure 5. The comparative convergence behavior for different alpha values

We noted from the above results that, GA is very sensitive for overall optimization degrees. For different degree of optimism (alpha), GA can give different objective value all of the times. Thus by giving different alpha value, GA would be able to give decision maker solution for the problem. The above results also show that GA can convergence to the best solution very fast.

5. Conclusion

The contribution of this paper consists of introducing the model of shortest path problem with fuzzy coefficient (fSPP) and providing GA-based algorithm to solve it. Here, the proposed method use the fuzzy ranking strategy based on the alpha value/degree of optimism. The performance of the proposed method was evaluated by using modified several test problems given in the literature. The findings reported in this study show that, GA is very sensitive for overall degrees of optimism. Thus, the decision makers can have their solution based on their degree of optimism. Moreover, the results also show that GA can convergence to the best solution very fast (less than 50 generations).

Acknowledgments

The author acknowledge The Directorate of Higher Education, The Ministry of Research, Technology, and Higher Education, Republic of Indonesia, for financial support through research grant “Hibah Bersaing” 2015.

References

[1] Bazaraa M S, Jarvis J J and Sherali H D 1990 *Linear Programming and Network Flows*, 2nd ed. (New York: John Wiley and Sons)

[2] Gen M and Cheng R 1997 *Genetic Algorithm and Engineering Design*, 2nd ed. (New York: John Willey and Sons).

[3] Dijkstra E W 1959 A Note on Two Problems in Connection with Graph *Numer. Math* Vol 1 p. 269–271

[4] Evans J R and Minieka E 1992 *Optimization Algorithms for Networks and Graphs*. (New York: Marcel-Dekker)

[5] Ericsson M, Resende M G C and Pardalos P. M. 2002 A Genetic Algorithm for the Weight Setting Problem in OSPF Routing *J. Comb. Optim.* vol. 6 p. 299 - 333

[6] Li Y Z, Ida K and Gen M 1997 Improved Genetic Algorithm for Solving Multi-objective Solid Transportation Problem with Fuzzy Number *Japanese Sosiety for Fuzzy Theory and System*, vol. 9, no. 2, p. 239–250

[7] Holland J 1975 *Adaptation in Natural and Artificial Systems* (Michigan: University of Michigan Press)

[8] Michalewicz Z 1994 *Genetic Algorithms+Data Structures = Evolution Programs* Spring-Verlag.

[9] Gen M and Cheng R 2000 *Genetic ALgorithm and Engineering Optimization* (New York: John Willey and Sons)

[10] Syarif A, Yun Y S and Gen M 2002 Study on Multi-stage Logistic Chain Network: A Spanning Tree-based Genetic Algorithm Approach *Comput. Ind. Eng.* vol. 43 no. 1 p. 299-314

[11] Syarif A and Gen M 2003 Solving Exclusionary Side Constraint Transportation Problem: A Genetic Algorithm with Fuzzy Logic Approach *J. Intell. Manuf.* vol. 14 p. 389-399

[12] Syarif A and Gen M 2003 Double Spanning Tree-Based Genetic Algorithm for Two Stage Transportation Problem *Int. J. Knowl. Based Eng. Syst. Des.*, vol. 7, no. 4, p. 214–221

[13] Syarif A, Aristoteles, Aryanti D and Riska M 2016 Performance Evaluation of Various Genetic Algorithm (GA) Approaches For Knapsack Problem *ARPN J. Eng. Appl. Sci.*, vol. 11, no. 7, p.

[14] Liou T and Wang M J 1992 Ranking Fuzzy Numbers with Integral Value *Fuzzy Sets Syst.*, vol. 50, no. 3, p. 247–255

[15] Ezzati R, Enayati R, Mottaghi A and Saneifard T R 2011 A New Method for Ranking Fuzzy

Numbers without Concernig of Real Numbers *J. Pure Appl. Math.* vol. 2, no. 2, p. 256–270

[16] Goldberg D and Lingle R 1985 Alleles, loci and the Traveling Salesman Problem, in *Proc. of the First International Conference on Genetic Algorithm*, p. 154–159

[17] Syswerda G 1989 Uniform Crossover in Genetic Algorithms, in *Proc of the 3rd International. Conference on Genetic Algorithm,* Morgan Koufman p. 2-9

[18] Gen M, Lin L and Cheng R 2004 Bicriteria Network Optimization Problem using Priority-based Genetic Algorithm *IEEJ Trans. Electron. Information. Syst.* vol. 124, no. 10, p. 1972–1978

[19] Ahn C W and Ramakrishna R 2002 A Genetic Algorithm for Shortest Path Routing Problem and the Sizing of Populations *IEEE Trans. Evol. Comput* vol. 6, no. 6, p. 566–579

[20] Hashier T and Munakata J D 1993 A Genetic Algorithm Applied to the Maximum Flow problem in *Proc. of the 5th Int. Conf. on Genetic Algorithms*, p. 488–493.

Appendix: Test Problems

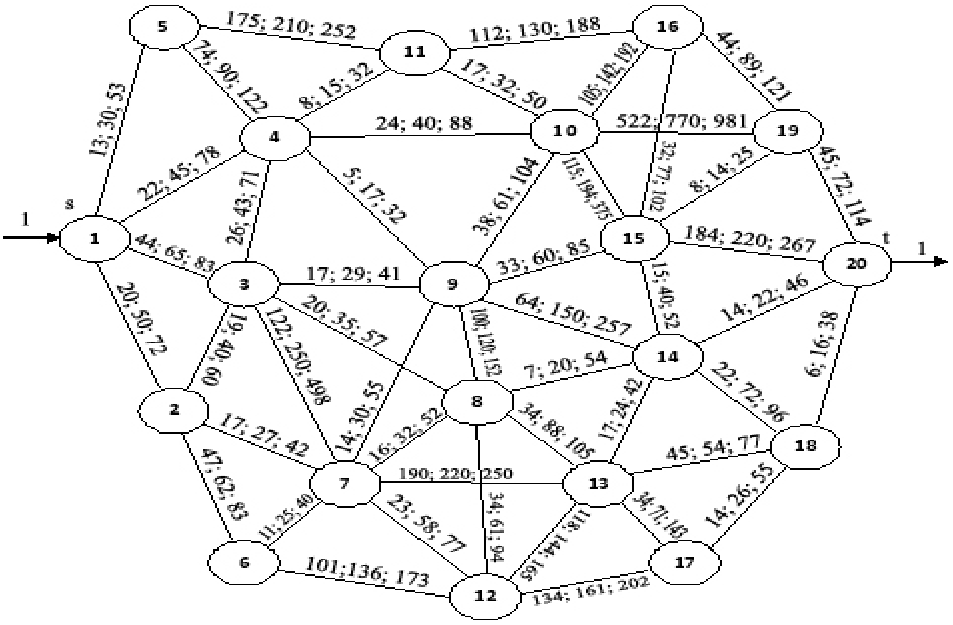


Figure 6. Test Problem 1 (modified from [19])

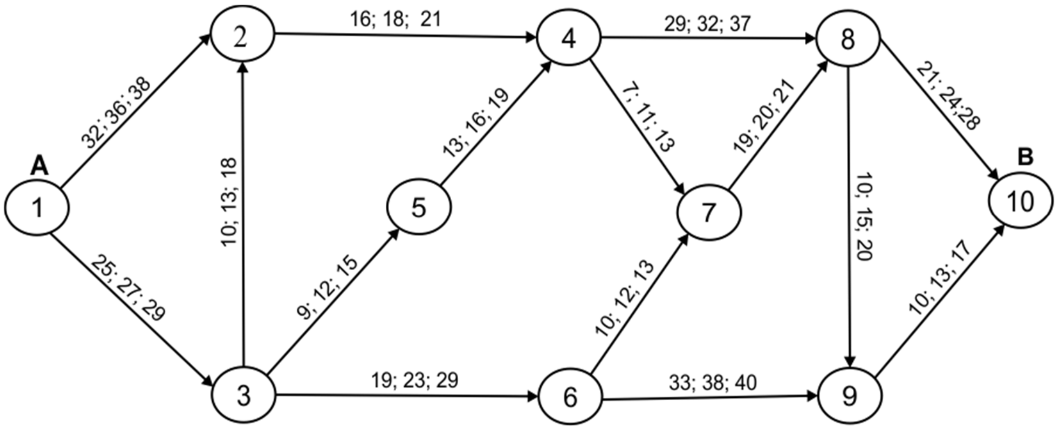


Figure 7 Test Problem 2 (modified from [2])

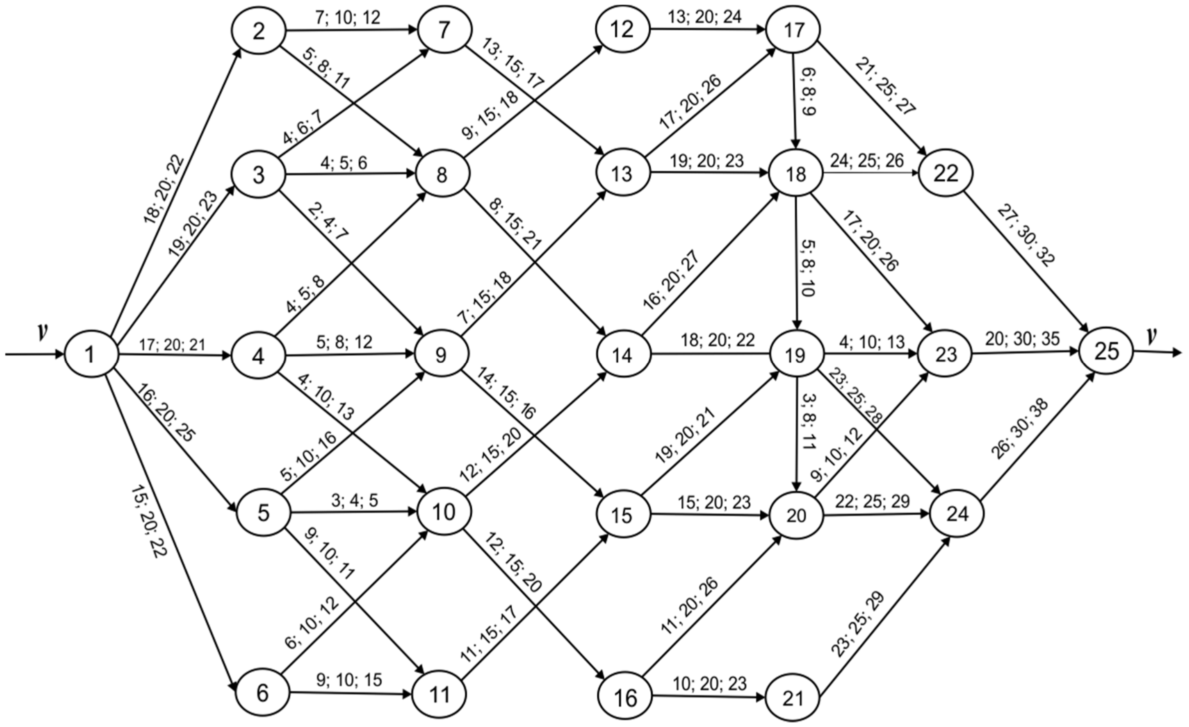


Figure 8 Test Problem 3 (modified from [20])