Interval Algorithms for Solving Minimal Spanning Tree and Shortest-Route Models

G. Gatev, A. Hossain

Key Words: *Interval minimal spanning tree; interval shortest-route; interval possibility; interval most reliable route; interval algorithms.*

Abstract. *Simple algorithms are presented for solving network models under parametric uncertainty. The new algorithms are applicable to the case when the generalized distance or the probability associated with each arc is nonnegative, interval, or real. The first three interval algorithms are developed on the base of midpoint and half-width representation of intervals, and these are more efficient than the interval algorithms that could be obtained by using traditional interval description. The fourth algorithm utilises the concept of interval possibility, and the interval product operator. The applicability of the result is demonstrated by considering several examples.*

1. Introduction

Network models have played an increasinly by important role in management science/operations research for at least two reasons. First, network models of real-world systems are relatively easy to conceive and construct. Second, network models can be communicated effectively to management as visual facsimiles of the real-world system under consideration.

Network models in operations research have evolved from the more general theory of graphs. In network terminology, the minimal spanning tree problem involves using the network branches to reach all nodes of the network in such a fashion that the length of all connecting branches be minimal. The minimal spanning tree algorithm is a simple one, but it has proven to solve one of the effective network models [15, 20, 26], and [29]. The shortest-route problem is concerned with determining the shortest-route from origin to a destination through a connecting network, given nonnegative distance associated with the respective arcs of the network [20, 26], and [29]. The most reliable route algorithm maximizes the probability of not being stopped on the route [29].

One of the earliest works in the area of graph or network models is the paper of Dijkstra [6]. Assuming n nodes, and the existence of at least one path between any two nodes, the author has considered two fundamental problems: to obtain the tree of minimum total length between the n nodes, and to find the path of minimum total length between two given nodes.

A variant of Minimum Spanning Tree Problem has been discussed in [2]. An $O(n^2)$ algorithm is proposed for determining a point on a given line l, which, if added to a given set of n nodes located on one side of l, yields the minimum spanning tree. To reduce complexity of the problem a divide-and- conquer technique is applied.

Two types of indirect covering tree problems have been introduced in [16], using a spanning tree as a backbone network, and these are the minimum cost covering subtree (MCCS) and the maximal indirect covering subtree (MICS). The objective of the MCCS is to find the minimum cost subtree in which all nodes are within a prescribed distance to a node of the subtree. Reduction techniques that have been used to solve the location set covering problem are extended to solve MCCS. MICS chooses that subtree which maximizes the demand within a distance standard of nodes of the subtree.

The quickest path problem, that arises when the transmission of data between two nodes of a network is considered has been treated in [25]. The problems of ranking the K quickest paths, the Chen's algorithm, and the Ranking K quickest loopless paths have been reviewed. The authors have also compared the algorithms in terms of the worst-case complexity.

In [21], an algorithm for solving a multi-criteria version of the shortest-route problem has been proposed. The distance is given by a multi-component vector and shortest is interpreted in the sense of vector minimum.

Two polynomial shortest path algorithms have been proposed in [13], for finding the shortest path from one node to all other nodes in a network. The algorithms are members of the family of Partitioning Shortest Path algorithms, and are based upon the threshold concept for partitioning scan eligible nodes. A set of shortest path models have been developed to accommodate the various applications.

In [1], two types of Bicriterion Shortest Path Algorithms have been discussed, based on path/tree approach and node labeling approach. The author has classified the different solution methods, and a ranking of the procedures based on the algorithmic structure has been suggested.

In the last two decades time windows constraints became an efficient way to model opening hours, preferred delivery time, etc. in many scheduling and routing problems. The shortest path problem with time windows consists of finding the least cost route between a source and a sink in a network, while respecting specified time windows at each visited node. A survey on the results and an efficient generalized permanent labeling algorithm to solve this problem in pseudo-polynomial time have been proposed in [22].

In [4], the authors have formulated the maximum covering/shortest path problem (MCSP) and provided solution experience for the maximum population/shortest path problem, a special case of the MCSP problem. The MCSP problem was formulated to analyze path options in terms of two conflicting objectives, namely, total path length, and total demand satisfied. The demand at a node is considered satisfied if the node is covered. A node is considered covered if either it is directly on the path or if it is within a predetermined maximum covering distance, S, from a node on the path.

A new algorithm for the general shortest paths ranking problem have been proposed in [17], that uses the path deletion concept. In a path deletion K shortest paths algorithm a sequence $\{g_1, g_2, \ldots, g_K\}$ of networks is defined, such that g_1 is the given network and its k-th shortest path is easily determined from the shortest path in g_k.

In many practical cases, the parameters of the network models are not exactly known, they are uncertain. Often this type of parametric uncertainty is handled within the framework of probabilistic models. In [14], the authors have examined a specific shortest path problem in acyclic network, in which arc costs are unknown functions of certain environment variables at network nodes, and each of these variables evolves according to an independent Markov process. Several procedures have been used to determine which of the environment states at each node are green (the vehicle departs immediately) and which are red (the vehicle waits), based on successive approximations, policy iteration, and parametric linear programming methods.

Risk and uncertainty have attracted the attention of theoretical economists, psychologists, engineers, mathematicians, decision-makers as well as empiricists in these fields. A survey of recent research results which use laboratory methods to contribute to the understanding of risk and uncertainty in environments which are of particular interest to managerial decision-making have been given in [7].

Another way to deal with parametric uncertainty is based on the usage of intervals, that are given by lower and upper limits, within which the values of the parameters are expected to fall.

In [8], an interval algorithm for solving shortest-route and dynamic programming problems based on midpoint and half-width representation of intervals has been proposed. This general approach yields computationally effective algorithms and is applied in this paper to develop most of the algorithms.

An $O(n^2)$ algorithm to solve all-pairs shortest path problem on an interval graph each edge of which has unit length has been proposed in [28]. The authors have constructed a corresponding „neighborhood tree" and extensively used the characteristic of this tree to develop and prove the correctness of the algorithm.

In [23], a simple algorithm for solving the all pairs shortest path problem on an interval graph G has been developed. The interval representation of graph G are given by lower endpoints (l_i) and upper endpoints (u_i).

A parallel algorithm on interval graphs and extension to circular-arc graphs have been discussed in [5]. The authors have considered all-pairs shortest path query problem: Given the interval model of an unweighted interval graph of n vertices, build a data structure such that each query on the shortest path between any pair of vertices of the graph can be processed efficiently.

After the seminal work of Zadeh [31], many authors have discussed fuzzy logic as a tool to deal with uncertainty, see, e.g. [3], [19], [27], [32].

The key concept of possibility, its close connection with the concept of membership in a fuzzy set, and its important role in the representation of meaning in the management of uncertainty and in application of the fuzzy approach to decision analysis, have been developed and treated in [3, 19, 27, 31], and [32].

Recently, in [30] a number of tools to aid the representation and processing of uncertain information have been discussed. The author has introduced a method for combining probabilistic and possibilistic information.

When uncertainties are included in network models (in the form of interval or fuzzy parameters) the computational burden is considerably increased. New algorithms with reduced computational complexity are needed for solving such models.

The aim of this paper is to the develop interval algorithms for solving the Minimal Spanning Tree Problem, the Shortest-Route Problem and the Most Reliable Route Problem under parametric uncertainty. In the cases of the Minimal Spanning Tree and the Shortest-Route Problems it is assumed that the uncertainty about the length of the arcs is described by intervals, and the mean-value lemma [8] is applied to develop the algorithms. The concept of interval possibility is introduced as an extension of the fuzzy graph's concept of possibility. In this way, an Interval Fuzzy Network Model to describe the Most Reliable Route Problem is obtained, and an interval algorithm for solving the model is developed.

The paper is organized as follows. The interval analysis concepts and some fuzzy graphs concepts are discussed in theoretical preliminaries in the second section. Interval Algorithms for Minimal Spanning Tree and Shortest-Route Models, as well as numerical examples to illustrate the applicability of the algorithms are presented in the third section. The obtained results are discussed in the conclusion in section 4.

2. Theoretical Preliminaries

First the interval analysis concepts are introduced [18, 24].

Let R be an interval. We will denote its lower (left) endpoint by $\underline{r}$ and its upper (right) endpoint by $\overline{r}$, so that $R = [\underline{r}, \overline{r}]$

The set of all intervals will be denoted by $I(R)$. Let $R, S \in I(R)$ and let * denote any of the interval arithmetic operations, * = +, -, x, /. Then the set theory definition of the interval arithmetic operations is as follows:

(1) $R * S = \{r * s | \ r \in R, s \in S\}$.

It follows that the sum of $R = [\underline{r}, \overline{r}]$, $S = [\underline{s}, \overline{s}]$ denoted by $R+S$, is the interval $R+S=[\underline{r},\overline{r}]+[\underline{s},\overline{s}]=[\underline{r}+\underline{s},\overline{r}+\overline{s}]$

The product $R\times S$ is again an interval

$$R\times S=[\min\{\underline{r}\underline{s},\underline{r}\overline{s},\overline{r}\underline{s},\overline{r}\overline{s}\},\ \max\{\underline{r}\underline{s},\underline{r}\overline{s},\overline{r}\underline{s},\overline{r}\overline{s}\}].$$

For $R,S>0$ the definition reduces to

(2) $R\times S=[\underline{r}\underline{s},\overline{r}\overline{s}]$.

The half-width of an interval $R=[\underline{r},\overline{r}]$ is the real number, $w(R)=\frac{1}{2}(\overline{r}-\underline{r})$, and the midpoint of R is the real number, $m(R)=(\underline{r}+\overline{r})/2$.

Using the set inclusion relation $\subseteq$ and the relation $\le$, we can define the supremum-like and infimum-like intervals:

(3) $\sup(R,S)=[\sup(\underline{r},\underline{s}),\sup(\overline{r},\overline{s})]$

(4) $\inf(R,S)=[\inf(\underline{r},\underline{s}),\inf(\overline{r},\overline{s})]$.

To compare intervals the concept of metric ρ is introduced. For each R and S in $I(R)$ the distance ρ is defined by

(5) $\rho(R,S)=\frac{1}{2}\{|\underline{r}-\underline{s}|+|\overline{r}-\overline{s}|\}$

Now the intervals R and S can be compared. The following important results hold [8].

$R\le S$ if and only if

(6) $\rho(R,\inf(R,S))\le\rho(S,\inf(R,S))$.

In a similar way,

$R\ge S$ if and only if

(7) $\rho(R,\sup(R,S))\le\rho(S,\sup(R,S))$.

Two intervals R and S are said to be equivalent $R\sim S$ if the following condition holds:

(8) $\rho(R, \sup(R, S)) = \rho(S, \sup(R, S))$.

(9) $\rho(R, \inf(R, S)) = r(S, \inf(R, S))$.

It means that $|\underline{r} - \underline{s}| = |\overline{s} - \overline{r}|$, i.e., the midpoints of R and S coincide.

In practical cases when $R\sim S$ and one have to make a choice in the sense of $\le$, the condition (6) should be modified. We say that $R\le S$ if

(10) $\rho(R, \inf(R, S)) = \rho(S, \inf(R, S))$ and $\underline{r}\le\underline{s}$

or

(11) $\rho(R, \inf(R, S)) = \rho(S, \inf(R, S))$ and $\overline{r}\le\overline{s}$.

We use, further, the notation $R\le S$ in the usual sense, when $\underline{r}\le\underline{s}$ and $\overline{r}\le\overline{s}$, and in the case of inclusion, $R\subseteq S$, when $\rho(R,\inf(R,S))\le\rho(S,\inf(R,S))$.

The conditions (6) and (7) lead to the following result, as proven in [8].

Let $m(P)$ denote the midpoint of P, $m(P) = (\underline{p}+\overline{p})/2$. Then

(12) $R\le S$ if and only if $m(R)\le m(S)$.

Let $[m(R), \Delta(R)]$ denote the interval R, $R=[\underline{r},\overline{r}]$, where $m(R)=(\underline{r}+\overline{r})/2$ is the midpoint of R, and $\Delta(R) = (\underline{r}-\overline{r})/2$ is the half-width of R, so that

$R = [m(R) - \Delta(R), m(R) + \Delta(R)]$,

or, using the new notation

(13) $R = [m(R), \Delta(R)]$.

The following result is easily shown:

Let R, S, and $T\in I(R)$. Then $T = R + S$ if and only if

(14) $m(T) = m(R) + m(S)$.

(15) $\Delta(T) = \Delta(R) + \Delta(S)$.

Now we introduce some fuzzy graphs concepts [19].

We shall consider in the finite graph G, $G\subset E\times E$ a path from x_{i1} to x_{ir}, that is an ordered r-tuple $P=(x_{i1},x_{i2},\ldots,x_{ir})$ where $x_{ik}\in E$, $k=1,2,\ldots,r$ and with the condition

$$\forall(x_{i_k},x_{i_{k+1}}):\ \mu(x_{i_k},x_{i_{k+1}})>0,\ k=\overline{1,r-1}$$

Let $X\wedge Y$ denote the operator min (X, Y). With each path a value is associated by

(16) $$P(x_{i_1},\ldots,x_{i_r})= =\mu(x_{i_1},x_{i_2})\wedge\mu(x_{i_2},x_{i_3})\wedge\ldots\wedge\mu(x_{i_{r-1}},x_{i_r}).$$

Let $P(x_i;x_j)$ be the set of all paths between x_i and x_j.

(17) $$P(x_i;x_j)= \{p(x_i;x_j)=(x_{i_1}=x_i,x_{i_2},..,x_{i_r}=x_j)\mid x_{i_k}\in E,k=\overline{2,r-1}\}.$$

The strongest path $P^*(x_i,x_j)$ from x_i to x_j can be obtained

(18) $$P^*(x_i,x_j)= \bigvee_{P(x_i;x_j)} P(x_{i_1}=x_i,x_{i_2}..x_{i_{r-1}},x_{i_r}=x_j)$$

where $X\vee Y=\max\{X, Y\}$.

The value defined by (16) may be extended to operators other than $\wedge$ under the restriction that these considered have the properties of associativity and monotonicity. Such an operator is for example, the product operator '$\times$'(ordinary multiplication), for which

If $a,b\in[0,1]$, then $a\times b\le a\wedge b$.

3. Interval Algorithms for Minimal Spanning Tree and Shortest-Route Models

3.1. Interval Minimal Spanning Tree Algorithm

The minimal spanning tree algorithm starts with any node and joining it to the closest node in the network. The resulting two nodes form a connected set, C, with the remaining nodes comprising in the unconnected set, $\overline{C}$. Next, connect the node from the unconnected set that is closest to any node in the connected set. The process is repeated until the unconnected set becomes empty, see, e.g. [29].

Let D_{ij} = The interval distance between node i and node j, and $D_{ij}=[\underline{d}_{ij},\overline{d}_{ij}]$.

The computational complexity of a straightforward interval generalization of the algorithm described above is relatively high, because the comparison of intervals would be based on using infimum-like intervals and distances, and because the interval arithmetic operations are more complex than the traditional ones.

A simple interval algorithm can be developed using the interval representation (13), $R = [m(R), \Delta(R)]$, where $m(R)$ and $\Delta(R)$ are midpoint and half-width of the interval R, and the conditions (12), (14), and (15).

The interval algorithm for minimal spanning tree consists of the following generalized steps [11]:

***Step* 1**. Describe the network using interval notation with midpoint and half-width. Denote the set of all connected nodes by C, the set of unconnected nodes by $\overline{C}$, $\overline{C} = N / C$, where N is the set of all nodes.

Denote the starting node by *Stn* and set $C = \{Stn\}$ and $\overline{C} = N / C$.

Set $d_{kr} = M$, $M >> 0$, when it is impossible to connect directly nodes k and r.

***Step* 2**. Choose (arbitrarily) node 1 from the network as starting node that is, *Stn=1*. Find the unconnected node j^* that is nearest to node 1

$$d_{1j^*} = \min_{j \in \overline{C}} \{d_{1j}\}; \; d_{ij} = \frac{\underline{d}_{1j} + \overline{d}_{1j}}{2}.$$

Connect nodes i and j^*, and set $C = \{1, j^*\}$, $\overline{C} = N / C$.

***Step* 3**. Identify the unconnected node that is closest to a connected node, and then connect these two nodes. If there is a tie, arbitrarily choose between them. This is accomplished in the following ways:

Obtain

$$d_{r^* k^*} = \min_{r \in C, k \in \overline{C}} \{d_{rk}\}.$$

Connect nodes r^* and k^*.

Set $C_{new} \{C, k^*\}$, $\overline{C}_{new} = N / C_{new}$

$C = C_{new}$ and $\overline{C} = \overline{C}_{new}$.

***Step* 4**. Repeat step 3 until all nodes are connected.

***Step* 5**. Obtain the midpoint $m(L)$ of the interval length L of the minimal spanning tree by adding the midpoint of all connecting branches. Obtain the half-width $\Delta(L)$of the interval length of the tree by adding the half-width Δ_{ij} of all connecting branches.

Numerical Example

Let consider the network in *figure 1*. The parameters (values) along the branches give the costs (or generalized lengths) D_{ij} of establishing links between nodes i and j. The cost is uncertain and represented by upper and lower limits.

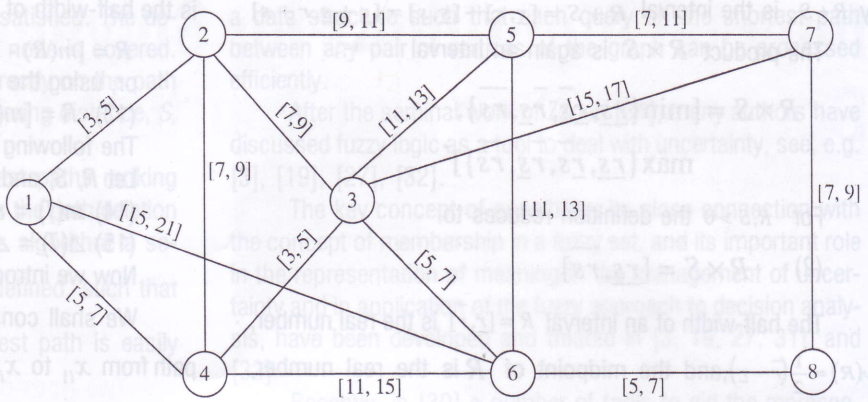


Figure 1

The graphical network of *figure 1* is presented in the *table 1*, using midpoint and half-width notation (step 1). M represents the case when there is no possible direct connection between nodes i and j, $M >> 0$.

Using the algorithm as described in section 3.1, the following computational results for minimal spanning tree problem are obtained and summarized in the *tables* below.

Tables 2 and 3 represent the results of iterations 1 and 2, respectively. In a similar way the results of iterations 3 to 7 can be obtained. After iteration 7, it is finally found that all the nodes have been connected. Our problem is now essentially solved. We need only look in *tables* and see which nodes are connected to give the solution of the minimal spanning tree problem. These are set out in *table 4*, from which it can be seen that the total cost (midpoint) is 43 units, and the total half-width is 8 units. The interval minimal spanning tree network is graphed in *figure 2*.

Table 1. Representation of the network in *figure1* using midpoint and half-width notation

From Nodes	To Nodes							
	1	2	3	4	5	6	7	8
1	[0, 0]	[4, 1]	M	[6, 1]	M	[18, 3]	M	M
2	[4, 1]	[0, 0]	[8, 1]	[8, 1]	[10, 1]	M	M	M
3	M	[8, 1]	[0, 0]	[4, 1]	[12, 1]	[6, 1]	[16, 1]	M
4	[6, 1]	[8, 1]	[4, 1]	[0, 0]	M	[13, 2]	M	M
5	M	[10, 1]	[12, 1]	M	[0, 0]	[12, 1]	[9, 2]	M
6	[18, 3]	M	[6, 1]	[13, 2]	[12, 1]	[0, 0]	M	[6, 1]
7	M	M	[16, 1]	M	[9, 2]	M	[0, 0]	[8, 1]
8	M	M	M	M	M	[6, 1]	[8, 1]	[0, 0]

Table 2. The result of step 2 iteration 1

From connected node	To Nodes							Minimum distance	Total connected nodes
	1	2	3	4	5	6	7		
1	[0, 0]	[4,1]	M	[6, 1]	M	[18, 3]	M	{4, 1}	{1, 2}
The new connected node is 2, the minimum distance is 4, and the half-width is 1.									

Table 3. The result of iteration 2

From connected nodes	To Nodes 1	2	3	4	5	6	7	8	Minimum distance	Minimum distance from *i* nodes	Total connected nodes *C*
1			M	[6, 1]	M	[18, 3]			[6, 1]	min= {[6, 1], [8, 1]}= [6, 1]	{1, 2, 4}
2			[8, 1]	[8, 1]	[10, 1]	M			[8, 1]		
The next connected node is 4, the minimum distance is 6, and the half-width is 1.											

We obtain the midpoint $m(L)$ and half-width $\Delta(L)$ of the minimal interval cost L

$m(L) = m_{1-2} + m_{1-4} + m_{4-3} + m_{3-6} + m_{6-8} + m_{8-7} + m_{7-5} = 4 + 6 + 4 + 6 + 6 + 8 + 9 = 43$

$\Delta(L) = \Delta_{1-2} + \Delta_{1-4} + \Delta_{4-3} + \Delta_{3-6} + \Delta_{6-8} + \Delta_{8-7} + \Delta_{7-5} = 1 + 1 + 1 + 1 + 1 + 1 + 2 = 8$

Hence, $L = [m(L), \Delta(L)] = [43, 8]$, and the minimal interval cost is obtained in the usual interval notation

$L = [\{m(L) - \Delta(L)\}, \{m(L) + \Delta(L)\}]$
$= [(43 - 8), (43 + 8)] = [35, 51]$.

The minimal interval cost $L = [35, 51]$

Table 4. The optimal solution of the minimal spanning tree problem

Iteration	Branch	Distance (midpoint)	Half-width
1	1-2	4	1
2	1-4	6	1
3	4-3	4	1
4	3-6	6	1
5	6-8	6	1
6	8-7	8	1
7	7-5	9	2
		Total distance= 43	Total half-width= 8

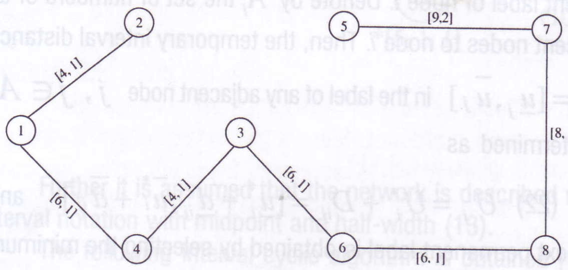


Figure 2. Optimal solution of the minimal spanning tree problem

3.2. Interval Shortest-Route Algorithms

The aim is to find the shortest-route between an origin node and a destination node in a network, given the nonnegative distance associated with the respective arcs of the network.

3.2.1. Interval Acyclic Algorithm

A network will be acyclic, if it does not have any loop. The acyclic algorithm is easier than the cyclic algorithm, because it yields fewer computations, see, e.g. [29].

Let D_{ij} and U_j denote the interval distance between nodes *i* and *j*, and the shortest interval distance from the source node (node 1) to node *j*, correspondingly. The destination node is node *n*.

The interval values of $U_j = [\underline{u}_j, \bar{u}_j]$, $j = \overline{2,n}$ may be computed recursively using the interval formula

(19) $U_j = \min_i \{U_i + D_{ij}\}$ where $U_i + D_{ij} = [\underline{u}_i + \underline{d}_{ij}, \bar{u}_i + \bar{d}_{ij}]$, and $U_1 = [0,0]$. The operator min{} is performed on the basis of the metric (5) and the conditions (6) or (10), (11). In this way an interval extension of the well-known acyclic algorithm is obtained.

We present a more effective algorithm, using the midpoint and half-width notation $R = [m(R), \Delta(R)]$ and the conditions (12), (14) and (15), see, [8].

Let u_j denote the real shortest distance from 1 to node *j*. The real values u_j, $j = \overline{2,n}$ are computed using the recursive noninterval formula

(20) $u_j = \min_i \{u_i + d_{ij}\}$

where d_{ij} is the midpoint of D_{ij}, $u_1 = 0$.

To obtain the optimal solution of the shortest-route problem, it is important to identify the nodes encountered along the route and the corresponding interval widths. The following labeling of node *j* is used

(21) node *j* Label = $[u_j, k, \Delta_{kj}]$

where *k* is the node immediately preceding *j* that leads to the shortest distance u_j and Δ_{kj} is the half-width of D_{kj}.

Further it is assumed that the network is described using interval notation with midpoint and half-width (13).

The generalized steps of the interval acyclic algorithm are summarized as follows:

Step 1. Assign the label [0,-,0] to source node.

Step 2. Compute the shortest distance from source node to the destination node *n*, by using the recursive formula (20). Label nodes by using (21).

***Step* 3.** Obtain the optimum route and the half-widths of the interval distance between nodes 1 and *n*, starting from node *n* and tracing backward through the nodes using the label's information.

***Step* 4.** To find the interval half-width of the solution add the corresponding Δ_{ij} encountered along the optimum route.

Numerical Example

Consider the network in *figure 3*. The generalized lengths of the arcs are uncertain and given by intervals, in the form (13).

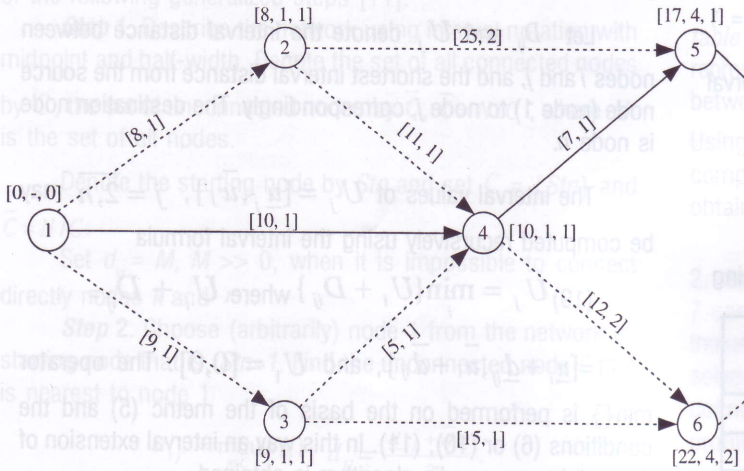


Figure 3

Using the algorithm, as described in section 3.2.1, we obtain the following results for nodes 1, 2, 3, 4, put on *table 5*. The computations for all iterations are summarized directly on *figure 3*.

Table 5

Node j	Computation of u_j	Connected from	Label
1	$u_1 = 0$	-	[0, -, 0]
2	$u_2 = 0{+}8 = 8$	node 1	[8, 1, 1]
3	$u_3 = 0{+}9 = 9$	node 1	[9, 1, 1]
4	$u_4 = \min\{0{+}10, 8{+}11, 9{+}5\} = 10$	node 1	[10, 1, 1]

The optimal solution is obtained tracing backward from node 7 and using the label's information.

7→[26,5,1]→(5,1)→[17,4,1]→(4,1)→[10,1,1]→(1,1).

The half-width of the optimal solution is

$\Delta_7 = \Delta_{57} + \Delta_{45} + \Delta_{14} = 1 + 1 + 1 = 3$.

Hence, $U_7 = [23, 29]$.

The algorithm provides the shortest interval distance between node 1 and any others node. In *figure 3*, the solid lines show the obtained shortest-route between the source and the destination node namely 1→4→5→7.

Note that if the interval formula (19) were used, at node 7, for example, we would have to compare two intervals [15, 19] + [8, 10] = [23, 29] and [19, 25] + [7, 9]= = [26, 34].

3.2.2. Interval Cyclic Algorithm

A network is said to be cyclic if it contains loops. The cyclic algorithm, known as Dijkstra algorithm, is more general in the sense that it subsumes the acyclic case. Temporary and permanent labels are used in the cyclic algorithm.

The source node is assigned a permanent label [0, -, -]. Then, we consider all adjacent nodes that have a direct connection from the last permanent node, and we determine their labels. These new labels are called temporary labels. The next permanently labeled node is chosen from among all temporary labeled nodes up to now as the one that has the minimum distance. This procedure is now repeated for the last permanently labeled node, see, e.g. [15], [26], and [29].

The temporary and permanent labels utilize the same format $[u, k]$, where u is the shortest distance found to date from the starting node to the corresponding node, and k is the number of the immediate predecessor node on that route.

A label status is converted to permanent if it has been ascertained that no shorter-route exists between the starting and the corresponding nodes. Otherwise, the label is temporary and may subsequently be updated.

Now we develop the interval version of Dijkstra's algorithm. Let $U_i^* = [\underline{u}_i^*, \overline{u}_i^*]$ be the interval distance in the permanent label of node *i*. Denote by A_i the set of numbers of all adjacent nodes to node *i*. Then, the temporary interval distance $U_j = [\underline{u}_j, \overline{u}_j]$ in the label of any adjacent node j, $j \in A_i$ is determined as

(22) $U_j = U_i^* + D_{ij} = [\underline{u}_i^* + \underline{d}_{ij}, \overline{u}_i^* + \overline{d}_{ij}]$ and

the next permanent label is obtained by selecting the minimum distance

(23) $U^* = U_k^* = \min_k \{U_k\}$, $k \in T$

using the definitions (4) – (7), (10), and (11). In (23) T represents the set of numbers of all temporary labeled nodes up to now, and $k^* = \arg(\min_{k \in T}\{U_k\})$.

A more effective algorithm is developed using the midpoint and half-width notation $R = [m(R), \Delta(R)]$, and the conditions

(12), (14), and (15).

Let u_j, u_i^* and d_{ij} denote the midpoints of the corresponding intervals U_j, U_i^* and D_{ij}. Then the interval formula (22) is replaced by a noninterval one

(24) $u_j = u_i^* + d_{ij}$, $u_1 = 0$, $j \in A_i$

and the next permanent label is obtained by comparing real (noninterval) values

(25) $u^* = u_k^* = \min_k\{u_k\}$, $k \in T$.

To obtain the optimal interval solution of the initial problem, it is important to identify the nodes and the half-width of intervals D_{ij} encountered along the route. This is achieved by using the following labeling:

(26) Node j label = $[u_j, k, \Delta_{kj}]$,

where u_j is the midpoint of interval U_j, and Δ_{kj} the is half-width of interval D_{kj}, and k is the last permanently labeled node.

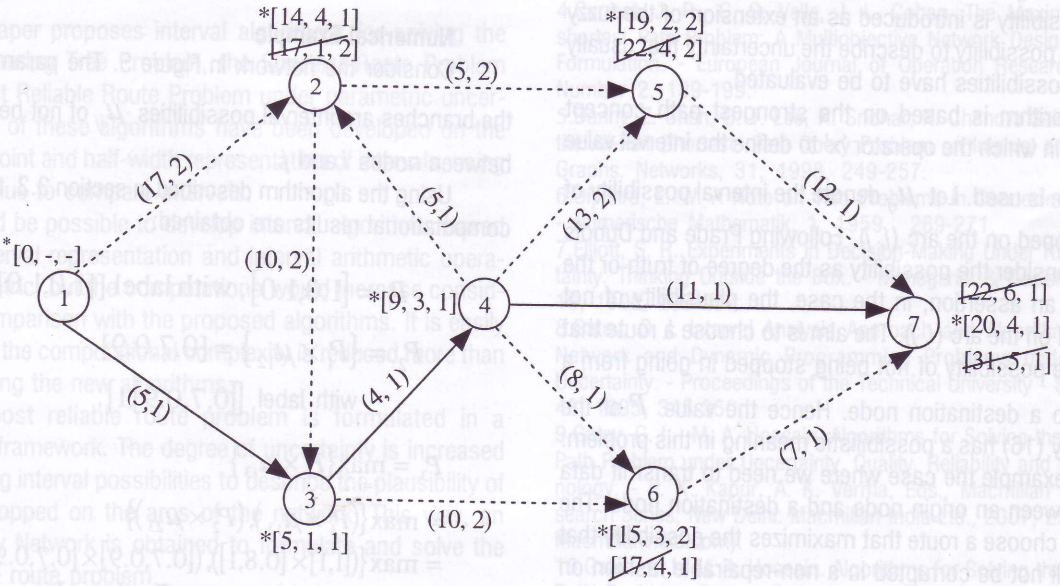


Figure 4

Further it is assumed that the network is described using interval notation with midpoint and half-width (13).

The following interval cyclic algorithm is obtained [12]:

***Step* 1.** Assign to node 1 the permanent label [0, -, -].

***Step* 2.** From the last permanently labeled node i, with label $[u_i, s, \Delta_{si}]$, obtain the temporary labels of all adjacent nodes j, $j \in A_i$. If an adjacent node is unlabeled, label it using (24). If the adjacent node is temporarily labeled $[u_j, s, \Delta_{sj}]$, $j \in A_i$, leave that label unchanged unless $u_i + d_{ij} \langle u_j$, in which case update the label, that is, change it to $[u_j = u_i + d_{ij}, i, \Delta_{ij}]$.

***Step* 3.** Consider the set $\{[u_j, k_j, \Delta_{k_j j}]\}$ of labels of all temporarily labeled nodes from iteration 1 to the current iteration, $j \in T$, and make permanent the label in which u_j is the smallest, $u^* = u_{j^*} = \min_{j \in T}\{u_j\}$ $j^* = \arg(\min_{j \in T}\{u_j\})$.

***Step* 4.** If all nodes are permanently labeled, the algorithm terminates otherwise, return to *Step* 2.

***Step* 5.** Obtain the optimum route between node 1 and the destination node n by tracing backward through the network using the label's information.

***Step* 6.** Add the corresponding half-width Δ_{ij} encountered along the optimum route using the label's information, and obtain the shortest interval distance between nodes 1 and n.

Numerical Example

Consider the network in *figure 4*. The values D_{ij} along the arcs represent generalized lengths (lengths, costs, or time), and are given by intervals, in the form (13).

Using the algorithm, we obtain the following results:

***Iteration* 0.** Assign the first permanent label [0, -, -] to node 1.

***Iteration* 1.** Nodes 2 and 3 can be reached directly from the last permanently labeled node 1, and the temporary labels are [17, 1, 2], [5, 1, 1] respectively.

The smallest distance u corresponds to node 3. Thus, node 3 is permanently labeled.

***Iteration* 2.** Nodes 4 and 6 have a direct connection with the last permanently labeled node 3, and the temporary labels are [9, 3, 1], [15, 3, 2].

Now, we have three temporary labels [17, 1, 2], [9, 3, 1], [15, 3, 2] associated with nodes 2, 4, 6, respectively. Node **4** has the smallest u = 9, hence its label [9, 3, 1] is changed to permanent, etc.

After iteration 6, all the nodes have permanent labels, thus the procedure is completed. The computational steps above are shown in *figure 4*. The shortest route is determined starting from node 7 and using label's information.

7→ [20, 4, 1] → (4, 1) → [9, 3, 1] → (3, 1)→ [5, 1, 1]→ (1, 1).

The shortest-route is 1 → 3 → 4 → 7.

The half-width of the optimal solution is obtained by summing the third elements in the labels.

$$\Delta_7 = \Delta_{47} + \Delta_{34} + \Delta_{13} = 1 + +1 = 3$$

Hence, $U_7 = [\underline{u}_7, \overline{u}_7] = [17, 23]$

Figure 4 shows the shortest-routes between node 1 and any node in the network, and the solid lines indicate the shortest-route from node 1 to node 7.

3.3. Interval Most Reliable Route Algorithm

The aim is to develop a simple algorithm for solving the most reliable route problem, when the possibilities of not being stopped on the segments of the route are uncertain. The concept of interval possibility is introduced as an extension of the fuzzy set concept of possibility to describe the uncertainty that usually exists when possibilities have to be evaluated.

The algorithm is based on the strongest path concept given by (18), in which the operator 'x' to define the interval value P of any route is used. Let μ_{ij} denote the interval possibility of not being stopped on the arc (i, j). Following Prade and Dubois (1996), we consider the possibility as the degree of truth or the plausibility of an assertion, in the case, the plausibility of not being stopped on the arc (i, j). The aim is to choose a route that maximizes the possibility of not being stopped in going from a origin node to a destination node. Hence the value P of the route, given by (16) has a possibilistic meaning in this problem. Consider for example the case where we need to transmit data packages between an origin node and a destination node. The problem is to choose a route that maximizes the possibility that a package will not be corrupted in a non-repairable fashion on the route. We shall refer to such similar situations as situations in which one wishes to maximize possibility of not being stopped on the route.

Let P_j = Interval possibility (generalized length) from node 1 to node j, and $P_j = [\underline{p}_j, \overline{p}_j]$. By definition for the starting node 1, $P_1 = [1.0, 1.0]$. The destination node is denoted by n.

The interval value of $P_j, j = \overline{1,n}$ will be computed recursively using the formula

$$P_j = \max_{i \in N_j}\{P_i \times \mu_{ij}\} \quad (27)$$

where i ranges over the set of all preceding nodes N_j. μ_{ij} is the interval possibility between current node j and its predecessor i, and $\mu_{ij} = [\underline{\mu}_{ij}, \overline{\mu}_{ij}], i \in N_j$.

To obtain the optimal solution of the problem, we will use the following label of node j:

$$\text{Node } j \text{ Label} = \{[\underline{p}_j, \overline{p}_j], b\} \quad (28)$$

where b is the node immediately preceding j, which yields the maximum P_j.

The interval most reliable route algorithm consists of the following generalized steps [9], [10]:

Step 1. Set j = 1. Assign to the source node (node 1) the label [[1.0, 1.0], -].

Step 2. Set $j = j + 1$. Compute the possibility P_j to node j using the formula (27). Label the node j by using the labeling (28).

Step 3. If $j = n$ go to step 4, else go to step 2.

Step 4. Obtain the optimum route between nodes 1 and n by tracing backward from node n through the nodes using label's information.

Numerical Example

Consider the network in Figure 5. The parameters along the branches are interval possibilities μ_{ij} of not being stopped between nodes i and j.

Using the algorithm described in section 3.3, the following computational results are obtained:

$$P_1 = [1.0, 1.0], \text{ with label } [[1.0, 1.0], -];$$

$$P_2 = \{P_1 \times \mu_{12}\} = [0.7, 0.9]$$

with label $[[0.7, 0.9], 1]$;

$$P_3 = \max_{i=1,2}\{P_i \times \mu_{i3}\}$$
$$= \max\{(P_1 \times \mu_{13}), (P_2 \times \mu_{23})\}$$
$$= \max\{([1,1] \times [0.8,1]), ([0.7,0.9] \times [0.7,0.9])\}$$
$$= \max\{[0.8,1], [0.49, 0.81]\} = [0.8,1]$$

with label $[[0.8,1],1]$

In a similar way, the possibilities related to the remaining nodes are determined.

The optimal solution is obtained by using label's information:

9 → [[0.408, 0.760], 5] → 5 → [[0.48, 0.80], 3] → 3 → [[0.80, 1.00], 1] → → [[1, 1], -].

Hence, the most reliable route is 1 → 3 → 5 → 9 with the corresponding interval possibility [0.408, 0.760]. In *fgure 5*, the solid lines indicate the most reliable route from the source node to the destination node.

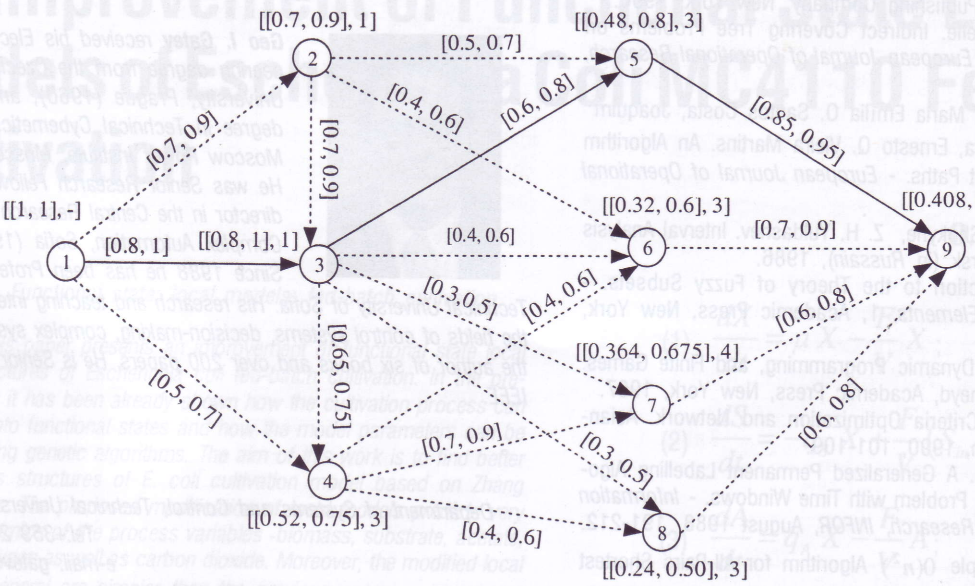


Figure 5

4. Conclusion

This paper proposes interval algorithms for solving the Minimal Spanning Tree Problem, the Shortest-Route Problem and the Most Reliable Route Problem under parametric uncertainty. Three of these algorithms have been developed on the base of midpoint and half-width representation of intervals, using the mean value to compare intervals.

It could be possible to develop interval algorithms based on usual interval representation and interval arithmetic operations and metric, but the computations would increase considerably by comparison with the proposed algorithms. It is easily realized, that the computational complexity is reduced more than twice, by using the new algorithms.

The most reliable route problem is formulated in a possibilistic framework. The degree of uncertainty is increased by introducing interval possibilities to describe the plausibility of not being stopped on the arcs of the network. This way, an Interval Fuzzy Network is obtained to formulate and solve the most reliable route problem.

These approaches yield simple and computationally effective algorithms for computing intervals that bound the sets of all solutions, when the exact values of the parameters of the network are unknown, but upper and lower limits within which the values are expected to fall are given. Numerical examples have been presented to illustrate the efficient assessment of the solution.

References

1.Anders, J. V. S. A Classification of Bicriterion Shortest Path (BSP) Algorithms. Asia-Pacific of Operational Research 2000,199-212.

2.Chen, G., G. Zhang. A Constrained Minimum Spanning Tree Problem. Computer & Operations Research, 2000, 867-875.

3.Chen, G.,T.T.Pham. Fuzzy Sets, Fuzzy Logic, and Fuzzy Control Systems. Introduction to, CRC Press, New York, 2001.

4.Current, J. R., C. S. Velle, J. L. Cohan. The Maximum Covering/ shortest Path Problem: A Multiobjective Network Design and Routing Formulation. - European Journal of Operation Research, 21, 1985, Number 2, 189-199.

5.Danny, Z. Chen, D. T. Lee, R. Sridhar, N. Chandra Sekharan. Solving the All-Pair Shortest Path Query Problem on Interval and Circular-Arc Graphs. Networks, 31, 1998, 249-257.

6.Dijkstra, E. W. A Note on Two Problems in Connexion with Graphs. - Numerische Mathematik, 1, 1959, 269-271.

7.Elliott, S. R. Experiments in Decision-Making Under Risk and Uncertainty: Thinking Outside the Box. - Managerial & Decision Economics 19, 1998, 239-257.

8.Gatev, G. I. Interval Analysis Approach and Algorithms for Solving Network and Dynamic Programming Problems under Parametric Uncertainty. - Proceedings of the Technical University - Sofia, Bulgaria, 48, 1995, 345-350.

9.Gatev, G. I. , M. A. Hossain. Algorithms for Solving the Most Reliable Path Problem under Uncertainty, Quality, Reliability and Infocom Technology. - P. K. Kapur, A. K. Verma, Eds., Macmillan Advanced Research Series, New Delhi: Macmillan India Ltd., 2007, 295-303 (www. Macmillanindia.com).

10.Gatev, G. I., M. A. Hossain. Algorithms for Solving the Most Reliable Path Problem. Proceeding of the 3rd International Conference on Engineering Education in Sustainable Development, Lyon, France, 4-6 Oct. 2006.

11.Gatev, G. I., M. A. Hossain. Algorithms for Solving Minimal Spanning Tree under Parametric Uncertainty. International IFAC Workshop, DECOM-TT 2004, Automatic Systems for Building the Infrastructure in Developing Countries, Regional and Global Aspects, Preprints, Oct. 3-5, 2004, Bansko, Bulgaria, 145-149.

12.Gatev, G. I., M. A. Hossain. Interval Shortest-Route Algorithms. International Conference. Automatics and Informatics'2000, Sofia, Bulgaria, 24-26 Oct., 2000, 68-71.

13.Glover, F., D. D. Klingman, N. V. Phillips, R. F. Schneider. New Polynomial Shortest Path Algorithms and their Computational Attributes. - *Management Science*, 31, Sept.1985,No. 9, 1106-1128.

14.Harilaos, N. P., N. T. John. Dynamic Shortest Paths in Acyclic Networks with Markovian Arc Costs. - *Operations Research Society of America*, 41, January-February, 1993, 91-101.

15.Hilier, S. F., G. J. Lieberman. Introduction to Operations Research. Fifth Edition, McGraw-Hill Publishing Company, New York, 1990.

16.Huston, V. A., C. ReVelle. Indirect Covering Tree Problems on Spanning Tree Networks. - *European Journal of Operational Research*, 1993, 20-32.

17.José Augusto Azevedo, Maria Emília O. Santos Costa, Joaquim João E.R. Silvestre Madeira, Ernesto Q. Vieira Martins. An Algorithm for the Ranking of Shortest Paths. - *European Journal of Operational Research*, 1993, 97-106.

18.Kalmikov, S. A., Yu. I. Shokyne, Z. H. Yuldashev. Interval Analysis Methods. Nauka, Novosibirsk (in *Russain*), 1986.

19.Kaufmann, A. Introduction to the Theory of Fuzzy Subsets. - *Fundamental Theoretical Elements*, 1, Academic Press, New York, 1975.

20.Kaufmann, A. Graphs, Dynamic Programming, and Finite Games. Translated by Henry C. Sneyd, Academic Press, New York, 1967.

21.Khan, Z. A. On Multi-Criteria Optimization and Network. Asian-Pacific Operation Research, 1990, 101-109.

22.Martin, D., S. Francois. A Generalized Permanent Labelling Algorithm for the Shortest Path Problem with Time Windows. - *Information Systems and Operational Research, INFOR*, August 1988, 191-212.

23.Mirchandani, P. A Simple $0(n^2)$ Algorithm for All-Pairs Shortest Path Problem on an Interval Graph. - *Networks*, 27, 1996, 215-217.

24.Moore, R. E. Interval Analysis. Prentice Hall, Englewood Cliffs, N. J., 1966.

25.Pascoal, M. M. B., M. E. V. Captivo, J. C. N. Climaco. - A Comprehensive Survey on the Quickest Path Problem. Ann Oper Res, 2006, 5-21.

26.Phillips, Don T., and A. Garcia-Diaz. Fundamentals of Network Analysis. Prentice-Hall, Inc., Englewood Cliffs, N. J., 1981.

27.Prade, H., D. Dubois. Fuzzy Logic, Implementation and Applications. Edited by Patyra, M. J., Mlynek, D. M., Wiley Teubner, New York, 1996.

28.Ravi, R., M. V. Marathe, C. Pandu Rangan. An Optimal Algorithm to Solve the All-pair Shortest Path Problem on Interval Graphs. *Networks* 22, 1992, 21-35.

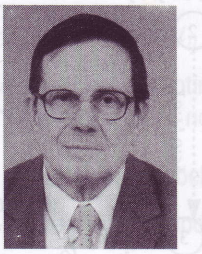
29.Taha, H. A. Operations Research, An Introduction. MacMillan Publishing Company, New York, 1992.

30.Yager, R. R. Fuzzy Set Methods for Uncertainty Management in Intelligence Analysis. - *International Journal of Intelligent Systems*, 21, 2006, 523- 544.

31.Zadeh, L. A. Fuzzy Sets. - *Information and control*, 8, 338- 353.

32.Zimmermann, H.-J. Fuzzy Set Theory - and its Applications. Second, Revised Edition, Kluwer Academic Publishers, Boston, 1991.

Manuscript received on 05.02.2007



Geo I. Gatev *received his Electrical Engineering degree from the Czech Technical University, Prague (1960), and his PhD degree in Technical Cybernetics from the Moscow Power Institute, Moscow (1968). He was Senior Research Fellow and vice-director in the Central Research Institute of Complex Automation, Sofia (1971-1986). Since 1988 he has been Professor at the Technical University of Sofia. His research and teaching interests are in the fields of control systems, decision-making, complex systems. He is the author of six books and over 200 papers. He is Senior Member of IEEE.*

Contacts:
Department of Systems and Control, Technical University of Sofia,
Tel:+359 2 965 25 96
e-mail: gatev@tu-sofia.bg

Akter Hossain *was born in Dhaka, Bangladesh. He received Master Engineering in Control Systems from the Technical University of Sofia, Bulgaria. He got specialization from the Technical University of Sofia. At the present time he is a PhD student at the International Doctorate School under French Department at Technical University of Sofia. Doing research in the field of decision-making and management, he is interested in network models, dynamic programming models under uncertainty.*

Contacts:
Ecole Doctorale Filiure Francophone Technical University of Sofia,
Tel. 00359 887317573
e-mail: akter_h@yahoo.com, akter@tu-sofia.bg