BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **9**, No 4

Sofia • 2009

Most Reliable Route Method and Algorithm Based on Interval Possibilities for a Cyclic Network

Akter Hossain

French Language Faculty of Electrical Engineering, Technical University of Sofia, 1117 Sofia
E-mail: akter_h@yahoo.com

Abstract**:** *A method and an algorithm are proposed for solving the most reliable route problem. The uncertainty about the reliability of a route is presented in a possibilistic setting. The concept of interval possibilities is introduced, as a generalization of the fuzzy sets concept of possibility, to deal with a higher degree of uncertainty.*

Keywords**:** *Networks, intervals, possibility, Most Reliable Route problem.*

1. Introduction

The paper is devoted to the development of an interval and fuzzy method and algorithm for solving the Most Reliable Route problem under parametric uncertainty. In the case when we need to transmit data packages between a source node and a destination node in a communication network and we have to select a route that links these two nodes, and maximizes the plausibility that a package will not be corrupted in a non repairable fashion on the route. Another example is associated with the transportation problem, when one has to determine a route that maximizes the possibility of not being stopped on the route.

The most reliable route problem is formulated in a probabilistic setting and solved using the well known shortest route algorithm [10]. In this paper, we consider higher degree of uncertainty. First, we use possibilities to represent the plausibility of not being stopped on the route. Thereafter, the degree of uncertainty is further increased by introducing the concept of interval possibilities as an extension of fuzzy sets concept of possibility [1]. Interval possibilities are more appropriate in the case, when the values of possibilities are uncertain, but expected to fall within given intervals.

A method to solve the most reliable route problem is proposed in [10]. The author converted probability to log probability. Then the classical shortest route algorithm is used to find the shortest distance (log). Finally, convert back this log probability to non-log probability.

Several simple methods and algorithms are proposed for solving the most reliable route problem in a finite fuzzy network under parametric uncertainty in [3, 4, 6]. The aim of these algorithms is to find the most reliable route in a given network that maximizes the possibility of not being stopped on the route. The possibilities on the route segments are certain and described by real values, or uncertain and described by an interval, given by upper and lower limits. These algorithms are applicable when the given network is acyclic and cyclic. The author analyzed the complexity of these algorithms, and all algorithms are polynomial algorithms [6].

Theoretical Preliminaries are given in Section 2. In Section 3, the most reliable route method and algorithm are described, the complexity of the algorithm is analyzed, and an example is considered. Conclusion is made in Section 4.

2. Theoretical preliminaries

First the interval analysis concepts are introduced [7, 9]. Let R be the set of all real numbers. By an interval V we mean a closed bounded compact subset of R:

$$V=\{v:v\in R,\ y\le v\le z,\ y,\ z\in R,\ -\infty<y\le z<\infty\}. \tag{1}$$

The set of all intervals is denoted by I (R). The real numbers of R are usually denoted by *small letters*, and the intervals of $I(R)$ are denoted by *capital letters*. Denote the *lower* (*left*) *endpoint* of interval V by $\underline{v}$ and its *upper* (*right*) *endpoint* of interval V by $\overline{v}$. Then the interval V is written as

$$V=[\underline{v},\ \overline{v}]=\{v|\underline{v}\le v\le \overline{v}\}.$$

Using the set inclusion relation $\subset$ and the relation $\le$, we can define the supremum-like (supr) and infimum-like (inf) elements [7]:

$$\text{supr}\,(V,\ W) = [\text{supr}(\underline{v},\ \underline{w}),\ \text{supr}(\overline{v},\ \overline{w})], \tag{2}$$

$$\inf\,(V,\ W) = [\inf(\underline{v},\ \underline{w}),\ \inf(\overline{v},\ \overline{w})], \tag{3}$$

where V, $W \in I(R)$, so that the partially ordered set $I(R)$ satisfies the structure of a lattice.

Let '$*$' denote any of the interval arithmetic operations, $+$, $-$, $\times$, $/$. Then the set theory definition of an interval arithmetic operation is as follows:

$$V*W=\{v*w|\,v\in V, w\in W\}.$$

The sum of $V=[\underline{v},\ \overline{v}]$, and $W=[\underline{w},\ \overline{w}]$ denoted by $V+W$, is again an interval, i.e., $Z=V+W\in I(R)$. Hence

$$V+W=[\underline{v},\overline{v}]+[\underline{w},\overline{w}]=[\underline{v}+\underline{w},\ \overline{v}+\overline{w}]. \tag{4}$$

To compare intervals the concept of metrics ρ is introduced. The distance $\rho(V, W)$ between two intervals $V = [\underline{v},\ \overline{v}]$, $W = [\underline{w},\ \overline{w}]$, and $V, W \in I(R)$, is defined by

(5) $$\rho(V\ ,\ W)\ =\ \frac{1}{2}\{|\,\underline{v} - \underline{w}\,| + |\,\overline{v} - \overline{w}\,|\}.$$

The intervals $V = [\underline{v},\ \overline{v}]$, $W = [\underline{w},\ \overline{w}]$, and $V, W \in I(R)$ can be compared. The following important result holds [2]:

$$V \le W \text{ iff}$$

(6) $$\rho(V,\ \inf(V,\ W)) \le \rho(W,\ \inf(V,\ W)).$$

In a similar way it can be proven that

$$V \ge W \text{ iff}$$

(7) $$\rho(V,\ \sup r(V,\ W)) \le \rho(W,\ \sup r(V,\ W))$$

and that

$$V \sim W \text{ (equivalence) iff}$$

(8) $$\rho(V,\ \sup r(V,\ W)) = \rho(W,\ \sup r(V,\ W)) = \rho(V,\ \inf(V,\ W)) = \rho(W,\ \inf(V,\ W)),$$

where $V \sim W$ means $V \subset W$ and $|\,\underline{v} - \underline{w}\,| = |\,\overline{v} - \overline{w}\,|$, i.e., the midpoints of V and W coincide.

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

(9) $$\rho(V,\ \inf(V,\ W)) = \rho(W,\ \inf(V,\ W)) \wedge \underline{v} \le \underline{w}$$

or

(10) $$\rho(V,\ \inf(V,\ W)) = \rho(W,\ \inf(V,\ W)) \wedge \overline{v} \le \overline{w}.$$

The choice among (9) and (10) depends on the application. When we need to obtain the smallest possible value, we may select (9). When we need to guarantee that the highest possible value be as small as possible, we may select (10).

We use, further, the notation $V \le W$ in the usual sense, when $\underline{v} \le \underline{w}$ and $\overline{v} \le \overline{w}$, and in the case of inclusion $V \subset W$, when $\rho(V,\ \inf(V,\ W)) \le \rho(W,\ \inf(V,\ W))$, or when (9) and (10) may be applied.

Let $m(V)$ denotes the midpoint of V, $m(V) = (\underline{v} + \overline{v})/2$. Then

(11) $$V \le W \text{ iff } m(V) \le m(W).$$

Denote by $[m(V),\ \Delta(V)]$ the interval V, $V = [\underline{v},\ \overline{v}]$, where $m(V) = (\underline{v} + \overline{v})/2$ is the midpoints of V, and $\Delta(V) = (\overline{v} - \underline{v})/2$ is the half-width of V, so that

(12) $$V = [\underline{v},\ \overline{v}]\ =\ [m(V) - \Delta(V),\ m(V) + \Delta(V)]$$

or

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

The following result is easily shown.

Let $V, W, Y \in I(R)$. Then $Y = V + W$ iff

(14) $$m(Y) = m(V) + m(W),$$

(15) $$\Delta(Y) = \Delta(V) + \Delta(W).$$

We will introduce some fuzzy networks concept [8]. Consider a finite graph G, $G \subset N \times N$. A route in a fuzzy network is a sequence of distinct nodes, x_{i_1}, x_{i_2}, $\ldots, x_{i_r}$, where $x_{i_k} \in N$, $k = 1, 2, \ldots, r$, with the condition

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

where $\mu_{\Re}(x_{i_k},\ x_{i_{k+1}})$ is the membership function of the pair ordered pair (x_{i_k}, $x_{i_{k+1}}$) for fuzzy relation $x_{i_k}\ \Re\ x_{i_{k+1}}$.

Let $X \wedge Y$ denotes the operator min (X, Y). With each route (x_{i_1}, x_{i_2}, $\ldots$, x_{i_r}) a value is associated by

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

Let $H(x_i;\ x_j)$ be the set of all ordinary routes between two arbitrary elements of N, x_i and x_j,

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

The strongest route $H^*(x_i,\ x_j)$ from x_i to x_j can be obtained

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

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. If $*$ is such an operator, then

(19) $$l(x_{i_1}, \ldots, x_{i_r}) = \mu_{\Re}(x_{i_1},\ x_{i_2}) * \mu_{\Re}(x_{i_2},\ x_{i_3}) * \ldots * \mu_{\Re}(x_{i_{r-1}},\ x_{i_r}).$$

In particular, if $*$ is a product operator, denoted multiplication (or '$\times$' ordinary multiplication) and defined in [8], then

(20) $$l(x_{i_1}, \ldots, x_{i_r}) = \mu_{\Re}(x_{i_1},\ x_{i_2}) \cdot \mu_{\Re}(x_{i_2},\ x_{i_3}) \cdot \ldots \cdot \mu_{\Re}(x_{i_{r-1}},\ x_{i_r}).$$

Due to the property

(21) $$v \cdot w \le v \wedge w \quad \text{if} \quad v, w \in [0, 1].$$

3. Most Reliable Route method and an algorithm based on interval possibilities for cyclic network

The aim is to develop a simple method and 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 method and the algorithm are based on the strongest route concept given by (18). Let μ_{ij} denotes the interval possibilities of not being stopped on the arc (i, j). Following Dubois and Prade (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 (source node) to a destination node. Hence the value l 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 be the interval possibilities (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 t, $t \le n$. The interval values of P_j, j = 2, …, t, will be computed recursively using the formula

(22) $$P_j = \max_{i \in N_j} \{P_i \times \pi_{ij}\},$$

where i ranges over the set of all preceding nodes N_j. π_{ij} is the interval possibilities between current node j and its predecessor i, and $\pi_{ij} = [\underline{\pi}_{ij}, \overline{\pi}_{ij}]$, $\underline{\pi}_{ij} \ge 0 \;\; i \in N_j$.

The formula (22) is represented in the form

(23) $$\log P_j = \max_{i \in N_j} \{\log P_i + \log \pi_{ij}\},$$

where, if F denotes any of the intervals P_j, P_i or π_{ij}, $\log F = [\log \underline{F}, \log \overline{F}]$. Since $\log \pi_{ij} \le 0$, and $\log P_i \le 0$, maximizing this sum is equivalent to the following minimization problem:

(24) $$U_j = \min_{i \in N_j} \{U_i + D_{ij}\},$$

where

(25) $$U_j = -\log P_j,\; U_i = -\log P_i,\; D_{ij} = -\log \pi_{ij}.$$

The operation min {} is performed on the basis of (6) and/or (9), (10).

Now we develop an algorithm based on interval possibilities for the cyclic network. Let U_i^* = $[\underline{u}_i^*, \overline{u}_i^*]$ be the interval distance of the last permanently labeled 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

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

and the label of node j is updated if the new distance U_j is shorter than the labeled distance up to now. The comparison of distances in the two labels $[[\underline{u}_j, \overline{u}_j], k]$ and $[[\underline{u}_j, \overline{u}_j], i]$ is accomplished using the definition (5), equations (3), (6), and the conditions (9) and (10).

The next permanent label is obtained by selecting the minimum distance

(27) $$U^* = U_{k^*} = \min_k \{U_k\}, \ k \in Q,$$

using again the (3), (5), (6), (9), and (10). In (27), Q represents the set of numbers of all temporary labeled nodes up to now, and $k^* = \arg(\min_{k \in Q}\{U_k\})$.

An effective interval algorithm is developed, using the midpoint and half-width notation (13), and conditions (11), (12), (14), (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 (26) is replaced by a noninterval one

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

and the label of node j is updated by comparing real numbers.

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

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

We use the following labeling of node j:

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

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

Further it is assumed that the network is described using interval notation with midpoint and half-width (13). The problem is to find a shortest route from node 1 to any node t, $t \le n$.

Denote by B the counter of iterations, by Q the set of all temporarily labeled nodes up to now, and by A_i the set of all adjacent nodes j to node i.

Hence, a preliminary step includes the conversion of interval possibilities to log interval possibilities, using (25), and then, the conversion of the usual interval notation in (24) to midpoint and half-width notation, using (13).

The following algorithm based on interval possibility for the cyclic network is proposed:

Step 1. Assign a permanent label $[u_1, k, \Delta_{k1}]$ to node 1. Assign temporary labels $[u_j, k(j), \Delta_{k(j)j}]$ to node j, $j = 2, \ldots, n$.

Define the sets A_i of all adjacent node i, $i = 1, \ldots, n - 1$.

Set i=1, u_i=0, k = –, Δ_{ki} = –, $u_j = \infty$, $k(j)j = -$, $\Delta_{k(j)j} = -$ for j = 2, …, n.

Set $B = 1$.

Set $Q = A_1$.

Step 2. From the last permanently labeled node i with label $[u_i, k, \Delta_{ki}]$, obtain the temporary labels of all adjacent nodes j, $j \in A_i$.

Leave the temporary label $[u_j, k(j), \Delta_{k(j)j}]$ of node j unchanged unless $u_i+d_{ij}<u_j$, in which case update the label, that is, change it to

$$[u_j = u_i + d_{ij},\ k(j) = i,\ \Delta_{k(j)j} = \Delta_{ij}],\ j \in A_i.$$

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, $j \in Q$, and make permanent the label in which u_j is the smallest, $u_{j^*} = \min_{j \in Q}\{u_j\}$, $j^* = \arg(\min\{u_j\})$.

Set $k = k(j^*)$, $u_i = u_{j^*}$, $\Delta_{ki} = \Delta_{k(j^*)j^*}$.

Step 4. If $B = n-1$ go to Step 5, otherwise,

Set $B = B+1$. Set $Q = (Q \mid k) \cup A_{k(j^*)}$.

Set $A_i = A_{j^*}$ and return to Step 2.

Step 5. Obtain the optimum route H between node 1 and the destination node t, starting from node t and tracing backward through the nodes using the label's information.

Step 6. Obtain the half-width $\Delta(U_t)$ of the interval solution U_t by summing the corresponding Δ_{ij} encountered along the optimum route $H^\bullet$,

$$\Delta(U_t) = \sum_{(i,j)\in H^\bullet} \Delta_{ij}.$$

Step 7. Obtain the interval solution U_t,

$$U_t = [\, u_t - \Delta(U_t),\ u_t + \Delta(U_t)\,].$$

Step 8. Obtain the shortest route logarithmic interval length and convert back to non-logarithmic notation.

3.1. Analysis of the complexity of the most reliable route algorithm based on interval possibilities for cyclic network

Consider the network in Fig. 1, where N = {1, 2, 3, 4, 5, 6, 7}, and A = {(1, 2), (1, 3), (1, 4), …, (6, 7)}, and the last permanently labeled node 1, u_1^* = [0, −, 0].

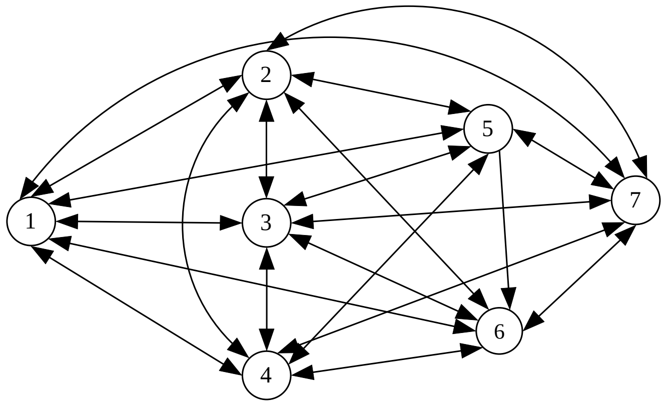

Fig. 1. Cyclic network with seven nodes

Iteration **1.** Nodes 2, 3, 4, 5, 6, 7 can be reached directly from the last permanently labeled node 1, u_1^* = [0, −, 0]. We need six additions to obtain the temporary labels $u_{i,1}$ corresponding to node 1, $u_{i,1} = u_1^* + d_{1i}$, i = 2, …, n, five comparisons are needed to find the next permanent label node, $\min\limits_{i=2,7}(u_{i,1})$ = $\min(u_{2,1}, u_{3,1}, u_{4,1}, u_{5,1}, u_{6,1}, u_{7,1})$. Assume the next permanently labeled node is 4, that is, $\min\limits_{i=2,7}(u_{i,1}) = u_{4,1}^*$.

In general, when the number of nodes is n, we need n−1 additions and n−2 comparisons.

Iteration **2.** Nodes 2, 3, 5, 6, and 7 can be reached from last permanently labeled node 4. We need five additions to compute the new temporarily labels with distances $u_{i,4}$, $u_{i,4} = u_{4,1}^* + d_{4i}$, i = 2, 3, 5, 6, 7. To find next permanently labeled node 9 comparisons are needed, $\min(u_{2,4}, u_{3,4}, u_{5,4}, u_{6,4}, u_{7,4}, u_{2,1}, u_{3,1}, u_{5,1}, u_{6,1}, u_{7,1})$. Assume the next permanently labeled is node 3 (from node 4), that is, $u_{3,4}^*$.

In general, when the number of nodes is n, we need n − 2 additions and $2(n − 3) + 1$ comparisons.

Iteration **3.** Nodes 2, 5, 6, and 7 have direct connection from the last permanently labeled node 3. Four additions are needed to compute the new temporary labels u_{i3}, $u_{i,3} = u_{3,4}^* + d_{3i}$, i = 2, 5, 6, 7. Eleven comparisons are needed to find the next permanently labeled node. Assume the next permanently labeled is node 5 (from node 1), that is, $u_{5,1}^*$.

In general, when the number of nodes is n, we need n − 3 additions and $3(n − 4) + 2$ comparisons.

Iteration **4.** Nodes 2, 6 and 7 can be reached directly from the last permanently labeled node, node 5. Three additions are needed to compute the new temporary labels $u_{i,5}$, $u_{i,5} = u^*_{5,1} + d_{5i}$, i = 2, 6, 7. Eleven comparisons are needed to obtain the next permanently label node. Assume the next permanently labeled is node 2 (from node 3), that is, $u^*_{2,3}$.

In general, when the number of nodes is n, we need $n - 4$ additions and $4(n - 5) + 3$ comparisons.

***Iteration* 5 or *Iteration* (n − 2).** Nodes 6 and 7 can be reached directly from the last permanently labeled node 2. Only two additions are needed to compute the new temporary labels, $u_{i,2}$, $u_{i,2} = u^*_{2,3} + d_{2i}$, i = 6, 7. Nine comparisons are needed to obtain the next permanently label node. Assume the next permanently labeled is node 6 (from node 3), that is, $u^*_{6,3}$.

In general, when the number of nodes is n, we need $n - 5$ additions and $5(n - 6) + 4$ comparisons.

Hence the number of additions is $n - (n - 2)$, and the comparisons is $(n - 2)(n - (n - 1)) + (n - 3)$.

***Iteration* 6 or *Iteration* (n − 1).** Node 7 has direct connection with the last permanently node 6. We need only one addition to get the new temporary label, $u_{7,6}$, $u_{7,6} = u^*_{6,3} + d_{67}$ and only five comparison to find the next permanently labeled node. Assume the next permanently labeled is node 7 (from node 5), that is, $u^*_{7,5}$.

In general, when the number of nodes is n, we need $n - (n - 1)$ additions and $6(n - 7) + 5$, or $(n - 1)(n - n) + (n - 2) = (n - 2)$ comparisons.

We have (for the general case):

Additions

$$(n - 1) + (n - 2) + (n - 3) + (n - 4) + \dots + n - (n - 2) + (n - (n - 1)) = \frac{n(n-1)}{2}.$$

Comparisons

$$(n - 2) + 2(n - 3) + 1 + 3(n - 4) + 2 + 4(n - 5) + 3 + \dots + (n - 2)(n - (n - 1)) +$$
$$+ (n - 3) + (n - 1)(n - n) + (n - 2) = \underline{A} + \underline{B} + \underline{C},$$

where

$$\underline{A} = n(1 + 2 + 3 + 4 + \dots + n - 2 + n - 1) = n\sum_{i=1}^{n-1} i = \frac{n^2(n-1)}{2},$$

$$\underline{C} = (1 + 2 + 3 + 4 + \dots + n - 3 + n - 2) = \sum_{i=1}^{n-2} i = \frac{(n-2)(n-1)}{2},$$

$\underline{B} = -2 - 2\times3 - 3\times4 - 4\times5 \dots -(n-2)\times(n-1) - (n-1)\times n \Rightarrow$ we may neglect it.

We need maximum n–1 additions to obtain half-width and another two additions to get traditional interval representation of shortest route logarithmic length plus two additions convert to non-logarithmic notation of shortest route length.

The total number of additions, $\underline{S}$, is as follows:

$$\underline{S} = \frac{n(n-1)}{2} + (n-1) + 2 + 2.$$

So, the ***running time*** of the algorithm is limited by O(addi = $\underline{S}$, comp = $\underline{A}$ +$\underline{C}$).

If we develop a most reliable route algorithm (based on interval possibility for cyclic network) based on traditional interval representation, the complexity of the algorithm will be very high (to compare two intervals **1** to **21** comparisons and **0** to **4** additions are needed, see, e.g. [6]).

3.2. Numerical example of the most reliable route algorithm based on interval possibilities for cyclic network

Consider the network represented in Table 1. Using (25) and (13), the network in Fig. 2 is obtained. As an example, $\pi_{12} = [\underline{\pi}_{12}, \ \overline{\pi}_{12}]$ = [0.7, 0.9] so that $\log \underline{\pi}_{12} = -0.1549$, $\log \overline{\pi}_{12} = -0.0458$ and [–1.0, –1.0] × [$\log \underline{\pi}_{12}$, $\log \overline{\pi}_{12}$] = [0.0458, 0.1549]. Using the midpoint and half-width notation (13), we obtain the equivalent representation of the interval D_{12}, $D_{12} = -\log \pi_{12} = [0.1004, 0.0546]$.

Table 1. Logarithmic transformation of interval possibilities

Road segment (i, j)	$\pi_{ij} = [\underline{\pi}_{ij}, \overline{\pi}_{ij}]$	$\log \underline{\pi}_{ij}$	$\log \overline{\pi}_{ij}$	$[-1,-1]\times[\log \underline{\pi}_{ij}, \log \overline{\pi}_{ij}]$	$[m(\log \pi), \Delta(\log \pi)]$
(1, 2)	[0.70, 0.90]	–0.1549	–0.0458	[0.0458, 0.1549]	[0.1004, 0.0546]
(1, 3)	[0.80, 1.00]	–0.0969	0.0000	[0.0000, 0.0969]	[0.0485, 0.0485]
(1, 4)	[0.50, 0.70]	–0.3010	–0.1549	[0.1549, 0.3010]	[0.2280, 0.0730]
(2, 3)	[0.70, 0.90]	–0.1549	–0.0458	[0.0458, 0.1549]	[0.1004, 0.0546]
(2, 5)	[0.50, 0.70]	–0.3010	–0.1549	[0.1549, 0.3010]	[0.2280, 0.0730]
(2, 6)	[0.40, 0.60]	–0.3979	–0.2218	[0.2218, 0.3979]	[0.3100, 0.0880]
(3, 4)	[0.65, 0.75]	–0.1870	–0.1249	[0.1249, 0.1870]	[0.1560, 0.0310]
(3, 5)	[0.60, 0.80]	–0.2218	–0.0969	[0.0969, 0.2218]	[0.1590, 0.0625]
(3, 6)	[0.40, 0.60]	–0.3979	–0.2218	[0.2218, 0.3979]	[0.3100, 0.0880]
(3, 8)	[0.30, 0.50]	–0.5228	–0.3010	[0.3010, 0.5228]	[0.4119, 0.1109]
(4, 6)	[0.40, 0.60]	–0.3979	–0.2218	[0.2218, 0.3979]	[0.3100, 0.0880]
(4, 7)	[0.70, 0.90]	–0.1549	–0.0458	[0.0458, 0.1549]	[0.1004, 0.0546]
(4, 8)	[0.40, 0.60]	–0.3979	–0.2218	[0.2218, 0.3979]	[0.3100, 0.0880]
(5, 9)	[0.85, 0.95]	–0.0705	–0.0222	[0.0222, 0.0705]	[0.0464, 0.0242]
(6, 9)	[0.70, 0.90]	–0.1549	–0.0458	[0.0458, 0.1549]	[0.1004, 0.0546]
(7, 3)	[0.30, 0.50]	–0.5228	–0.3010	[0.3010, 0.5228]	[0.4119, 0.1109]
(7, 9)	[0.60, 0.80]	–0.2218	–0.0969	[0.0969, 0.2218]	[0.1590, 0.0625]
(8, 9)	[0.60, 0.80]	–0.2218	–0.0969	[0.0969, 0.2218]	[0.1590, 0.0625]

Using this algorithm we obtain the following results:

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

Iteration 1. Nodes 2, 3 and 4 can be reached directly from the last permanently labeled node 1, and the temporary labels are [0.1004, 1, 0.0546], [0.0485, 1, 0.0485], and [0.228, 1, 0.073] respectively.

The smallest distance u corresponds to node 3. Thus, node **3** is permanently labeled, with label **[0.0485, 1, 0.0485]**.

Iteration 2. Nodes 4, 5, 6 and 8 have direct connection with the last permanently labeled node 3, and the temporary labels are [0.2045, 3, 0.031], [0.2075, 3, 0.0625], [0.3585, 3, 0.088], [0.4604, 3, 0.1109].

Node 4 has two temporary labels {[0.228, 1, 0.073], [0.2045, 3, 0.031]}, we will keep the new temporary label, because it includes the smallest distance.

Now, we have five temporary labels [0.1004, 1, 0.0546], [0.2045, 3, 0.031], [0.2075, 3, 0.0625], [0.3585, 3, 0.088] and [0.4604, 3, 0.1109] associated with nodes 2, 4, 5, 6 and 8 respectively. Node **2** has the smallest $u = u^*$ = min (0.1004, 0.2045, 0.2075, 0.3585, 0.4604) = 0.1004, hence its label **[0.1004, 1, 0.0546]** is changed to permanent, and so on.

All the computational results are summarized in Fig. 2.

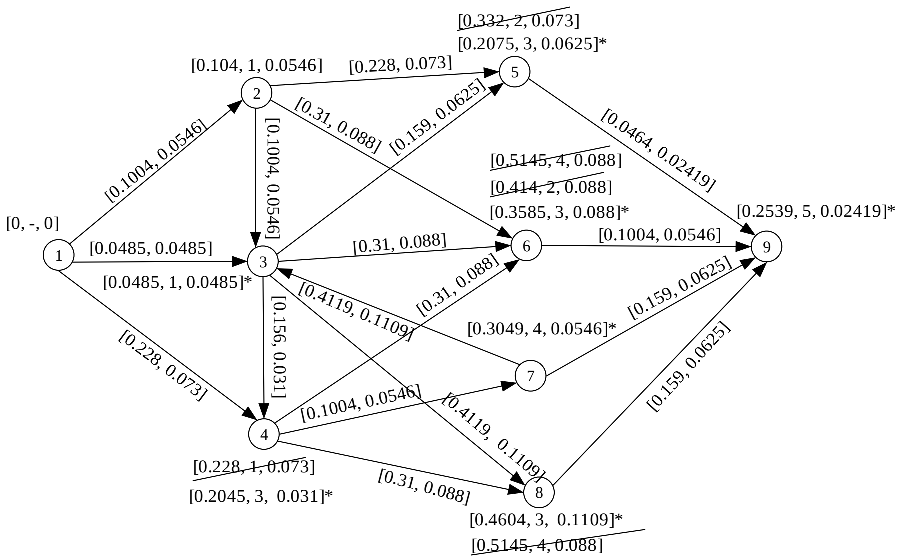

Fig. 2. Most reliable route based on interval possibilities for cyclic network

Tracing backward using label's information the most reliable route is as follows:

$$9 \Rightarrow [0.2539, 5, 0.02419] \Rightarrow 5 \Rightarrow [0.2075, 3, 0.0625] \Rightarrow 3 \Rightarrow [0.0485, 1, 0.0485]] \Rightarrow 1 \Rightarrow [0, -, 0].$$

The half-width of the optimal solution is: $\Delta_9 = \Delta_{59} + \Delta_{35} + \Delta_{13} =$ **0.1352**.

We have got the midpoint and half-width values, and now it is possible to obtain the interval possibility.

$$U_9 = [\underline{u}_9, \overline{u}_9] = [(0.2539 - 0.1352), (0.2539 + 0.1352)] = [0.1187, 0.3891].$$

$$U_j = -\log P_j.$$

Thus, $\log P_9 = [-0.3891, -0.1187]$. So, P_9 = **[0.408, 0.760]**.

The most reliable route is (**1 ⇒ 3 ⇒ 5 ⇒ 9**), and the corresponding interval possibility is **[0.408, 0.760]**.

4. Conclusions

A method and an algorithm are proposed for solving the most reliable route problem in a finite fuzzy cyclic network. The uncertainty about the reliability of a route is represented in a possibilistic setting. The plausibility of not being stopped on a segment of the route is described using the corresponding possibilities. The concept of interval possibilities is introduced to increase the degree of uncertainty. The new algorithm maximizes the possibilities of not being stopped on the route between an origin node and a destination node. The complexity of the algorithm is evaluated.

The transformation of the initial representation into a logarithmic form is accomplished only once at the beginning and than the simple midpoint algorithm [5, 6] for solving the interval shortest-route problem is applied. The new approach yields a simple and computationally effective algorithm when the exact values of parameters are unknown, but the upper and lower limits within which the values are expected to fall are given. Instead of comparing intervals using distance (5) and supremum-like or infimum-like intervals (2) or (3), the algorithm compares real values, i.e., the midpoints of intervals. A numerical example is given to illustrate the efficient assessment of the solution and the workability of the developed algorithm.

R e f e r e n c e s

1. D u b o i s, D., H. P r a d e. Fuzzy Sets in Approximate Reasoning: A Personal View. – In: Fuzzy Logic, Implementation and Applications. M. J. Patyra, D. M. Mlynek, Eds. New York, Wiley Teubner, 1996, 3-32.
2. G a t e v, G. Interval Analysis Approach and Algorithms for Solving Network and Dynamic Programming Problems under Parametric Uncertainty. – In: Proc. of the Technical University of Sofia. Vol. **48**. 1995, 345-350.
3. G a t e v, G., A. H o s s a i n. Algorithms for Solving the Most Reliable Path Problem under Uncertainty. – In: Quality, Reliability and Infocom Technology. P. K. Kapur, A. K. Varma, Eds. Macmillan Advanced Research Series. Macmillan India Ltd., 2007, 295-303.
4. G a t e v, G., A. H o s s a i n. Algorithms for Solving the Most Reliable Path Problem. – In: Proc. of the 3rd International Conference on Engineering Education in Sustainable Development. Lyon, France, 4-6 October 2006 (Full Paper – Disc Version, EESD 2006, C. Subaï, D. Ferrer and K. Mulder, Eds. © INSA de Lyon, ENTPE, ENSMSE, CIRIDD, 2006).
5. G a t e v, G., A. H o s s a i n. Interval Shortest Route Algorithms. – In: Proc. of the International Conference on Automation & Information ‘2000. Vol. **2**, Sofia, Bulgaria, 24-26 October, 2000, 68-71.
6. H o s s a i n, A. Network Models under Parametric Uncertainty. PhD Thesis, Technical University of Sofia, 2009.
7. K a l m i k o v, S. A., Y u. I. S h o k y n e, Z. H. Y u l d a s h e v. Interval Analysis Methods. Novosibirsk, Nauka, 1986 (in Russain).
8. K a u f m a n n. A. Introduction to the Theory of Fuzzy Subsets, Fundamental Theoretical Elements. Vol. **1**. New York, Academic Press, 1975.
9. M o o r e, R. E. Interval Analysis. Englewood Cliffs, Prentice Hall, New Jersey, 1966.
10. T a h a, H. A. Operations Research. An Introduction. Eighth Edition. Delhi, Pearson Education Inc., 2007.