Some Problems of Fuzzy Modeling of Telecommunications Networks

Kirill Garbuzov  
Novosibirsk State University  
Department of Mechanics and Mathematics  
Novosibirsk, Russia, 630090,  
Email: gartesk@rambler.ru

Alexey S. Rodionov  
ICM&MG SB RAS  
Novosibirsk, Russia, 630090,  
Email: alrod@sscc.ru

Abstract—Some common ideas and problems concerning the use of fuzzy logic for modeling unreliable networks are discussed on simple example. Approach to measuring fuzziness of different fuzzy parameters types is presented along with rules of their mutual conversions. The difficulty of keeping information at these conversions is shown.

I. INTRODUCTION

Modern communication systems, sensor networks in particular, provide us with many problems concerning calculation and optimization of models corresponding to real infrastructure parts. This also includes some other kinds of networks, like transport.

Many of these models are representing unreliable networks, i.e. networks with failing components. The most common example is probabilistic models, where each component has probability of failure [1]–[3]. But this method has its own cons since we can’t always have sufficient statistical data or even have to use expert opinion. Alternative approach is using fuzzy models for description of reliability.

Fuzzy models help avoid problem of insufficient statistical data allowing membership function to have values between 0 and 1. Fuzzy logic was introduced in 1965 by Lotfi A. Zadeh. It has been applied to many fields involving descriptions of uncertain events.

This approach allows us to weaken the formalism of classic probabilistic models and simplify the solution of our problem. There is also a wide choice of ways of representing fuzzy parameters, which gives us flexibility to build completely different models for different purposes. In last years, we can find a lot of researches concerning usage of fuzzy models for network’s analysis and optimization [4]–[6]. In all these and other papers that consider fuzzy models of telecommunication networks, a unified approach for description of uncertainty is used, that is some unique model is chosen, for example triangle numbers or intervals. At the same time, when describing large network, different teams of experts may be recruited that use different techniques and so present results in different measures of uncertainty. For analysis of such mixed data conversion to some unique presentation is needed. This conversion may lead to loss of information thus increasing uncertainty of a result. In this paper we discuss the problem and propose some methods for conversion of uncertainty presentations with minimal loss of information.

II. INFORMATIVENESS OF NETWORK ELEMENTS

Using fuzzy logic in networks modeling allows us to formulate the problems, which can’t be considered in standard probabilistic approach [1]. Calculating of network components’ informativeness is one of these problems.

Let G be an arbitrary graph with unreliable edges, which are set by fuzzy numbers. Let’s call these parameters the possibility of presence of each edge. For each edge we can also introduce fuzziness - some function showing us how much information we can get from this edge. One of the examples is...
logarithmic entropy, which can be easily calculated for most cases of classic fuzzy numbers. At last, we can talk about informativeness - a value, which shows us how much influence has the change of fuzziness to the value of the objective function. In practice this problem could stay for the process of maintenance of big network in the state of lack of resources. Knowing the informativeness, we can consider, which elements of our network are more important in terms of specifying their condition.

In this paper in most of the examples we are using the logarithmic entropy as the measure of fuzziness:

\[
d(A) = \int_{U} S(\mu_A(x)) \, dx;
\]

where \( S(y) = -y \cdot \ln(y) - (1-y) \cdot \ln(1-y) \) - Shannon function.

Let’s consider a simple example to show the problem closely. Let each edge of \( G \) have a triangular number \( \langle \gamma, a, \delta \rangle : 0 < a < 1; a-\gamma \geq 0; a+\delta \leq 1 \) as a fuzzy parameter. Operations between them could be defined as:

\[
\langle \gamma_1, a_1, \delta_1 \rangle \ast \langle \gamma_2, a_2, \delta_2 \rangle = \langle \max(\gamma_1, \gamma_2), a_1 \ast a_2, \max(\delta_1, \delta_2) \rangle
\]

Here we can consider these triangular numbers as probabilities of presence for the graph elements with some additional parameters (\( \gamma \) and \( \delta \)).

In the case (b) we are calculating the possibility of connection of two subgraphs. Without changing parameters this value equals \( \langle 0.1, 0.8, 0.1 \rangle \). Alternately modifying fuzzy numbers corresponding to both edges of the bridge, we get different results. It means that these elements are not equal in terms of informativeness.

Consequently, informativeness of elements can be affected not only by their own fuzzy parameters but also by the structure of the whole network and parameters of other edges. It is obvious that we only need to compare the values of informativeness of elements, so we can try to use different heuristic algorithms.

We are currently checking the hypothesis, which lets us to use not the whole graph but only some subgraphs containing the edges we try to compare. This localization could dramatically decrease the calculation difficulty (e.g. using the minimal chain containing objective edges makes the solution trivial).

### III. Conversion of Fuzzy Numbers

Converting different types of fuzzy parameters is another important problem. If we get network parameters from different sources, they can be represented in different types of fuzzy numbers. So we have to convert them to one common type and keep the most information they initially have. The problem is: how to choose this common type to lose the least amount of informativeness?

Knowing the membership function of the fuzzy number we can easily find the logarithmic entropy. E.g. for triangular numbers \( \langle \gamma, a, \delta \rangle \) it equals \( \frac{1}{2}(\gamma + \delta) \). Now if we know the measure of informativeness we can set arithmetic operations to correspond to any specific problem (depending on the objective function, network structure etc.). So we have to commit not only the measure but also the set of operations on our fuzzy parameters.

The problem appears when we try to convert these parameters from one type to another e.g. trapezoidal numbers to triangular or triangular numbers to intervals. Let’s consider another simple example to show it.
Let $A$ be a triangular number $\langle \gamma, a, \delta \rangle$. As shown above, its logarithmic entropy equals $\frac{1}{2} (\gamma + \delta)$. Let’s build a trapezoidal number by changing the membership function to 1 on the interval with length $d_1 + d_2$ (Fig. 2(a)). Its entropy equals $\frac{1}{2} (\gamma + \delta - d_1 - d_2)$. As we see, logarithmic entropy decreases and this can be treated as information loss since we don’t use any additional information about network to change that parameter.

If we try to keep the value of one chosen fuzziness measure (Fig. 2(b)), we obviously get the changes in other ways of calculating it. So we have to choose this function carefully for the current problem in order to avoid the artificial changes of entropy.

Another important special case is representing fuzzy parameters by intervals, which actually can not be straightly represented as classic fuzzy numbers. Arithmetic of intervals is a special algebraic system, which formalizes operations on intervals. Converting classic fuzzy numbers to intervals and back can be done in different ways.

The easiest idea is to build membership function as constant on the interval with length $\gamma + \delta$. Entropy then equals:

$$d = (\gamma + \delta) (-h \cdot ln (h) - (1 - h) \cdot ln (1 - h)).$$

Keeping the entropy value we can get $h \approx 0.8$ or $h \approx 0.2$. But here we obviously lose information about the peak of the initial parameter.

To show the reverse connection between these types of fuzzy numbers we formulate the following lemma:

**Lemma 1:** Let the fuzziness measure on intervals and triangular numbers be continuous, strictly monotone functions $\phi (\gamma, \delta)$ and $\psi (l)$, and $\psi$ is differentiable on [0,1]. Now if for some interval with length $l = \gamma + \delta$ we have $\phi (\gamma, \delta) \leq \psi (\gamma + \delta)$, then additional fuzziness on converting interval to triangular number is bounded by the value depending only on $\psi$.

**Proof:** Let’s point out that we only need to proof the case, where the entropy growth, i.e.:

$$\psi (l) = \psi (\gamma + \delta) \leq \phi (\gamma', \delta').$$

Consequently from the condition of lemma:

$$\psi (\gamma + \delta) \leq \psi (\gamma' + \delta').$$

Then using the mean value theorem:

$$\Delta d \leq \psi (\gamma' + \delta') - \psi (\gamma + \delta) \leq \psi' (l + \epsilon (l' - l)) (l' - l) \leq \max_{x \in [0,1]} \psi' (x).$$

Remark: In lemma proof $l' - l$ is bounded by 1, but in practice this value can be much smaller. ■

We can define membership function for intervals as 0.5 on the interval with length $l$. It gives estimation for entropy difference ($\approx 0.2l$). Lemma gives rough estimation for reverse conversion and provides better results for special cases.

**IV. Algorithms Examples**

**A. Search of spanning trees**

Problem of searching the minimum spanning tree can be easily solved by many algorithms. Getting a single spanning tree is also a trivial task, which can be resolved by either depth-first search or breadth-first search in linear time. These trees can be used for calculating network connection probability.

For any graph, the number of spanning trees can be calculated using Kirchhoff’s matrix-tree theorem. In the fuzzy models we can try to find $k$ best spanning trees, or even trees with fuzzy restrictions. Difficulty of such problems can vary depending on our model. For example, we can represent the informal description of heuristic algorithm for searching the set of trees meeting the fuzzy restrictions on connection possibility. It is based on the local search method. On each iteration we try to change one of the edges of current tree without breaking the connection. As a result we get some set of spanning trees without the full search. Depending on the fuzziness measure and initial restrictions we can ignore “unpromising” trees during each iteration.
B. Connection of bipartite graph

Let G be a bipartite graph with a set of fuzzy edges as a bridge between its subgraphs. We need to choose a subset of these edges to have the maximal connection possibility with limited summary weight (represented by another fuzzy parameter in common case).

With additional restrictions we can get different problems of discrete optimization. For example, with prohibition on "boundary" nodes to have more then one incidental edge from the "bridge" set, we get the problem of maximal matching. Here we don’t have any additional restrictions, so we get the 0-1 knapsack problem. All initial parameters are considered fuzzy numbers.

Let’s look more carefully at the greedy algorithm. Like for the real parameters, it doesn’t always give the exact solution. For testing we used triangular numbers with two different sets of operations for weights and possibilities. Since the possibilities of presence must be limited by the fuzzy analogue of segment [0,1], operations were formulated in this way: centers were handled as usual probabilities and boundaries were found as mean values.

For weights we additionally had to define multiplicative inverse element and full order. So we used operations defining the field of triangular numbers:

\[ \langle \gamma_1, a_1, \delta_1 \rangle + \langle \gamma_2, a_2, \delta_2 \rangle = \langle \gamma_1 \cdot \gamma_2, a_1 + a_2, \delta_1 \cdot \delta_2 \rangle \]

\[ \langle \gamma_1, a_1, \delta_1 \rangle \cdot \langle \gamma_2, a_2, \delta_2 \rangle = \langle e^{\ln(\gamma_1) \cdot \ln(\gamma_2)}, a_1 \cdot a_2, e^{\ln(\delta_1) \cdot \ln(\delta_2)} \rangle \]

Multiplicative inverse value:

\[ \langle \gamma, a, \delta \rangle^{-1} = \langle e^{\frac{1}{\ln(\gamma)}}, \frac{1}{a}, e^{\frac{1}{\ln(\delta)}} \rangle \]

It is possible to avoid defining some additional operations on weight, e.g. by using any other function growing with possibility growth and decreasing with weight growth instead of common ratio. Algorithm is obviously polynomial: sorting for \( O(n \cdot \ln(n)) \) and then passing \( n \) elements.

1: function GREEDY(costs, weights, limit)
2: for \( i = 0..n - 1 \) do
3: \( \text{ratios}[i] = \text{costs}[i] \cdot \text{weights}[i]^{-1} \)
4: \( \text{resultVector}[i] = \text{false} \)
5: end for
6: \( \text{currSum} = 0 \)
7: \( \text{cost} = 0 \)
8: while \( \text{currSum} \leq \text{limit} \) do
9: \( \text{currMax} = 0 \)
10: \( \text{max} = -1 \)
11: for \( i = 0..n - 1 \) do
12: \( \text{tempVal} = \text{ratios}[i] \)
13: if \( \text{currMax} < \text{tempVal} \) and \( \text{resultVector}[i] \) then
14: \( \text{tempSum} = \text{currSum} + \text{weights}[i] \)
15: if \( \text{tempSum} \leq \text{limit} \) then
16: \( \text{currMax} = \text{tempVal} \)
17: \( \text{max} = i \)
18: end if
19: end if
20: end for
21: if \( \text{max} \geq 0 \) then
22: \( \text{currSum}+ = \text{weights}[\text{max}] \)
23: \( \text{cost}+ = \text{costs}[\text{max}] \)
24: \( \text{resultVector}[\text{max}] = \text{true} \)
25: else
26: break
27: end if
28: end while
29: return \( \text{cost} \)
30: end function

In this algorithm we consider all values template, extending FuzzyNumber class (i.e. triangular numbers). All used operations must be defined for the exact types of fuzzy numbers. This approach has to use the multiplicative inversion, which we don’t need for brute-force search. And most of the results for the normal greedy algorithm for knapsack problem can be applied in our case as well.

Having the trivial realization, this problem shows flexibility and convenience of fuzzy models. This algorithm was implemented using object-oriented programming language Java, as a part of a big package dedicated to fuzzy networks. First of all,
using interfaces allows to make a template algorithm without specifying type of fuzzy numbers and operations on them. Secondary, it is easy to think of hierarchy of fuzzy numbers, which can also be easily represented using object-oriented paradigm (Fig.3). Any algorithm can be implemented as a separate generic class, and interfaces allow to monitor the correctness of operations in use.

Fig. 3. Example of fuzzy numbers class hierarchy

V. Conclusion

The brief results presented above shows some problems in fuzzy modeling of unreliable telecommunication networks. We hope that the problem of using different presentations of uncertainty in description of one unreliable network will attract attention of other researchers. Our initial theoretical results for converting fuzzy parameters and informal descriptions of algorithms on fuzzy models may be useful for automation of constructing fuzzy models of telecommunication and other networks.

Our next goal is comparing network elements’ informativeness by using the “localization” method. We also want to check the advantages of switching between different types of fuzzy parameters on the examples of big networks with much fuzzy data represented in three or more ways.

VI. Acknowledgments

This work is supported by the grant of the Program of basic researches of the Presidium of Russian Academy of Science.

REFERENCES