Proceedings of the 2007 IEEE Symposium on
Foundations of Computational Intelligence (FOCI 2007)

Fuzzy Partial-Order Relations for Intervals and Interval Weighted Graphs

Ping Hu
Computer Science Department
Conway, AR 72035
phu@uca.edu
www.cs.uca.edu

Chenyi Hu
Computer Science Department
Conway, AR 72035
chu@uca.edu
www.cs.uca.edu

Abstract

Weighted graphs have been broadly employed in various kinds of applications. Weights associated with edges in a graph are constants mostly in the literature. However, in real world applications, these weights may vary within ranges rather than fixed values. To model such kind of uncertainty or variability, we propose interval-valued weighted graphs in this study.

In solving practical graph applications such as finding shortest paths and minimum spanning trees for interval weighted graphs, it is necessary to be able to compare interval valued weights. However, two general intervals can not be ordered reasonably in binary logic. In this paper, we establish fuzzy partial-order relations for intervals. These relations are continuous, except only at a single point in a special case. After studying the properties of the fuzzy partial order relations, we applied the interval partial order to extend well-known shortest path and minimum spanning tree algorithms for interval weighted graphs.

Keywords

Interval weighted graph, fuzzy partial-order relation for intervals, interval shortest path and minimum spanning tree

1. INTRODUCTION

1.1 A brief review of weighted graphs

In this paper, we study graphs whose weights are intervals instead of constants used in the literature. In order to do this, we briefly review fundamental concepts of graph theory here. There is a very rich amount of literature on graphs. The 'theory of graphs' named by J. Sylvester (1814-1897) began with Euler's paper (1735) describing the problem of the seven bridges of Königsberg and was further developed by A. Cayley (1821-1895), W. Hamilton (1805-1865) and others.

In general, a graph G consists of a set of vertices (V) and a set of edges (E), i.e. G = (V, E). If the edges in the graph do not have directions, G is an undirected graph. Otherwise, it is a directed graph or digraph. A path of a graph is a consecutive sequence of edges. G is connected if for any two vertices A and B in a graph G there exists a path in G such that one can travel between A and B. This is only a sufficient but unnecessary condition for connectivity if a graph is directed. G is weighted if for every $e \in E$ there is a weight w_e associated with e. These weights can represent meaningful things such as distance, cost, and others in applications. Therefore, weighted graphs have been well studied and broadly applied in solving real world applications. Graphs studied in this paper are initially assumed to be positive weighted, and connected, undirected as the sample in Figure 1. This assumption is purely for the simplicity of our discussion. As we will see later in this paper, results reported can be extended to digraphs even with negative weights (without negative cycles) as well.

Among typical applications of weighted graphs are finding shortest paths, minimum spanning trees, and others. Algorithms for finding shortest paths include Dijkstra's algorithm [8] (1959), Bellman-Ford algorithm [2] (1958) and [9] (1962), and others. Algorithms that find a minimum spanning tree include Kruskal's algorithm [16] (1956), Prim-Jarnik algorithm [19] (1957) and Borůvka's algorithm [5] (1926). All of these algorithms require the order relationship of real numbers to determine optimal solutions. Figure 1 below is a sample connected weighted graph with six vertices and eleven undirected edges. The shortest path from A to F is weighted six as $A \rightarrow C \rightarrow E \rightarrow F$. The edges AC, CB, CE, ED, and EF form the minimum spanning tree of the graph with total weight 12.

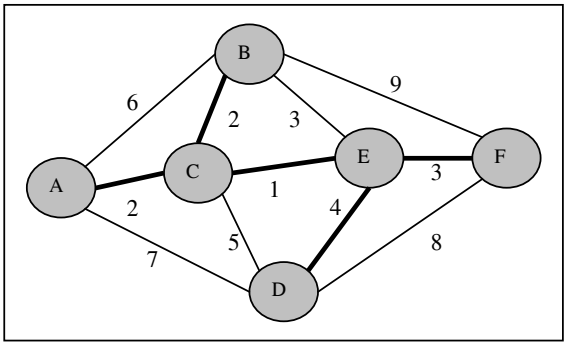

Figure 1: A weighted graph

1.2 Motivations of this study

We have noticed that in the current literature weights associated with edges are constants in a weighted graph. However, in real world applications, due to some kinds of uncertainties, weights associated with edges often vary within ranges rather than fixed constants. Here are few examples. Travel time (driving or flight) between A and B may not be exactly two hours but between an hour and 50 minutes and two hours five minutes mostly. The available bandwidth of a network connection may be 75-80% during a given time period. The price of a stock during a day can be between $8.08 -$8.88. To better model the variability of weights in a graph, instead of using constants, we represent weights as intervals. An interval [a, b] is the set of all real numbers between its

lower (left) and upper (right) bounds a and b provided that a, b$\in \Re$, and $a \le b$. When a = b, the interval is *trivial* and is the same as a real number. In the rest of this paper, we use boldfaced letters to represent intervals. The left and right endpoints of an interval are denoted with subscripts L and R. For example, $\mathbf{a} = [a_L, a_R]$ is an interval whose left and right endpoints are a_L and a_R, respectively.

The rest of this paper is organized as the follow. Section 2 defines interval weighted graph and a brief review of related concepts in interval computing. In section 3, we present a binary operator (and its dual) between two intervals; and then prove that the operator and its dual form fuzzy partial order relationships for intervals. Properties of the fuzzy partial order relations are also presented in the section. In sections 4 and 5, we apply the fuzzy partial order relationship to study algorithms that find shortest paths and minimum spanning trees for interval weighted graphs, respectively. We conclude this paper with section 6.

2. INTERVAL WEIGHTED GRAPHS AND INTERVAL COMPUTING

2. 1 Interval weighted graphs

As mentioned in the previous section, we study graphs with interval valued weights in this paper. We define the concept of interval weighted graph as the follow:

Definition 1: A graph G = {V, E} is an interval weighted graph if for each edge e ∈ E there is an interval weight $\mathbf{w}_e$ associated with it.

As an example, Figure 2 below is an interval weighted graph.

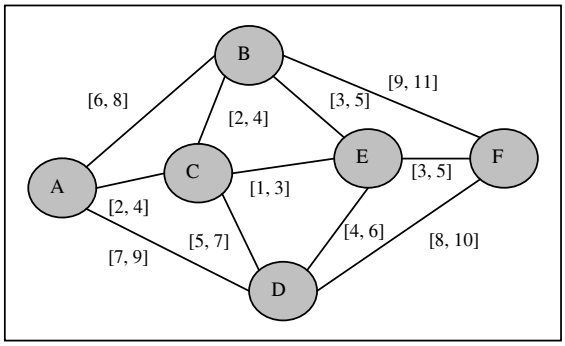

Figure 2: An interval weighted graph

Here we would like to clarify that the concept of *interval weighted graph* is completely different from the term *interval graph* in the existing literature.

The concept of interval graph is a type of intersection graph introduced by Benzer in [3] (1959). In 1964, Gilmore and Hoffman defined interval graph [11] as: "A graph G is an interval graph if and only if every quadrilateral in G has a diagonal and every odd cycle in G^c (G's complementary graph) has a triangular chord." In [10] (1985), Fishburn stated: "It [interval graph] refers a graph (X, ~) whose points can be mapped into intervals in a linearly ordered set such that, for all distinct x and y, x ~ y if and only if the intervals assigned to x and y have a nonempty intersection." In other words, interval graph is a special type of graph such that the orders of its vertices satisfy the above specified conditions but nothing involving the weights of the edges.

The term *interval weighted graph* that we have defined here means only that the weights associated with edges are interval valued. In other words, the order of vertices is not in the consideration of studying interval weighted graphs in this paper.

2.2 Interval arithmetic

To study interval weighted graphs, we need interval computing proposed by Moore [17] in 1950's. As mentioned previously in this paper, an interval is a set of real numbers defined as:

$$\mathbf{a} = [a_L, a_R] = \{x \in \Re \mid a_L \le x \le a_R\}.$$

The left and right endpoints a_L, a_R of the interval **a** on the real line are also called the lower and upper limits of the interval, respectively. We say that **a** is a non-negative interval if $a_L \ge 0$. In addition to the left-right endpoints representation, an interval **a** can also be represented by its midpoint m (**a**) together with its radius r (**a**), where $m(\mathbf{a}) = (a_L + a_R)/2$ and $r(\mathbf{a}) = (a_R - a_L)/2$. It is easy to see that $\mathbf{a} = [a_L, a_R] = [m(\mathbf{a}) - r(\mathbf{a}), m(\mathbf{a}) + r(\mathbf{a})]$. If the radius of **a** is zero, i.e. $a_L = a_R$, then **a** is a trivial interval the same as an ordinary real number.

Interval arithmetic has been defined by Moore [17] as an approach of putting bounds on rounding errors in mathematical computation and thus obtaining reliable results. Where classical arithmetic defines operations on numbers, interval arithmetic defines a set of operations on intervals. The basic binary arithmetic operations for two intervals, $\mathbf{a} = [a_L, a_R]$ and $\mathbf{b} = [b_L, b_R]$, are:

$\mathbf{a + b} = [a_L + b_L, a_R + b_R]$

$\mathbf{a - b} = [a_L - b_R, a_R - b_L]$

$\mathbf{a * b} = [\min(a_Lb_L, a_Lb_R, a_Rb_L, a_Rb_R), \max(a_Lb_L, a_Lb_R, a_Rb_L, a_Rb)]$

$\mathbf{a / b} = [\min(a_L/b_L, a_L/b_R, a_R/b_L, a_R/b_R), \max(a_L/b_L, a_L/b_R, a_R/b_L, a_R/b_R)]$ provided that $0 \notin \mathbf{b}$.

These arithmetic operations can be represented in terms of midpoints and radiuses of **a** and **b** as well. Here are simple examples of interval arithmetic operations:

Example 2: Let **a** = [1, 2] and **b** = [3, 4] be two intervals. Then:

$\mathbf{a + b} = [4, 6]$, $\mathbf{a - b} = [-3, -1]$,

$\mathbf{a * b} = [3, 8]$, and $\mathbf{a / b} = [1/4, 2/3]$.

There is a very large amount of literature on interval analysis far beyond the scoop of this paper. Interested readers may check the comprehensive website [14] maintained by Professor Kreinovich to find more information about interval computing.

3. A FUZZY PARTIAL-ORDER RELATIONSHIP FOR INTERVALS

3.1 Incomparability of intervals in binary logic

In studying shortest paths and spanning trees of a weighted graph, one needs an ordering relationship to compare distances and/or total weights. For any two given real numbers *x* and *y*, the statement '*x* is less than *y*' can be either true or false depending on their positions on the real line. The ordering relation of two real numbers can be presented perfectly in classical binary logic. However, for two nonempty intervals **a** and **b**, the relation '**a** is less than **b**' can be fairly complicated. In [1] (1983), Allen listed 13 possible casess for the temporal relationships of two time intervals without quantification. Instead of listing all of the 13 cases, we use Figures 3-5 to illustrate three of them.

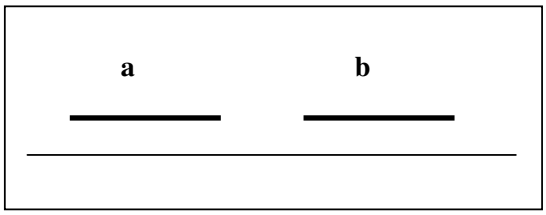

Figure 3: Disjoint intervals

In Figure 3, intervals **a** and **b** are disjoint. Since $\forall$ a $\in$ **a** and $\forall$ *b* $\in$ **b**, a < b, one can say that '**a** is less **b**' is true without any hesitations. For example, if **a** = [2, 4] and **b** = [5, 9] then '**a** is less than **b**'. This is what has been widely accepted in [1] and others.

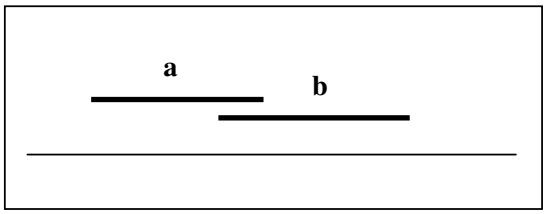

Figure 4: Partially overlapping intervals a and b

However, in Figures 4 and 5, there are some a $\in$ **a**, and b $\in$ **b**, such that a < b; and there are also some a' $\in$ **a**, and b' $\in$ **b**, such that a' < b'.

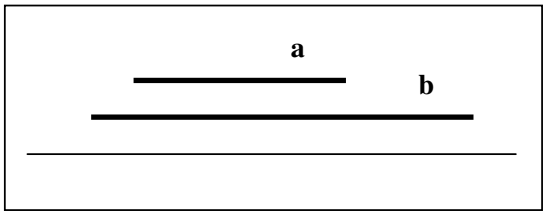

Figure 5: Nested intervals a and b

Therefore, the statement '**a** is less than **b**' cannot be simply represented in traditional binary logic anymore since it is true under a certain degree and false in another degree depending on their relative positions. It is fuzzy indeed.

Here we would also like to point it out that, in [10] (1985), Fishburn defined a concept of *interval order*. The concept was for a special kind of partially ordered set defined as: "A partially ordered set $(X, \prec)$ is an interval order precisely when its points x, y, … can be mapped into intervals in a linearly ordered set, such as $(\Re, <)$, such that, for all x and y in X, x $\prec$ y if and only if the interval assigned to x completely proceeds the interval assigned to y." From this definition we can clearly see that the concept of interval order is not for ordering intervals but for a special kind of partially ordered set. The purpose of reviewing this concept is for studying the order of vertices in interval graphs.

In studying temporal relationship of time intervals [1], Allen needs only qualitative relationship of two time intervals. In [18] Nguyen and et al investigated points in intervals via a probabilistic approach without ordering intervals. However, in studying interval weighted graphs, we need a quantitative relationship for intervals. After thorough searches on the internet, we have not found quantitative comparisons for intervals. Since intervals are incomparable in binary logic in general as we described previously, it calls for a fuzzy order relation for comparing intervals. We can then quantitatively compare two intervals with fuzzy memberships.

3.2 Fuzzy partial-order relations for intervals

In [24] (1965), Zadeh proposed fuzzy logic for statements that can be both true and false in certain degrees. He further defined the concept of fuzzy membership to quantitatively describe the degree of the belief.

In [6] (2006), Collins and Hu initially investigated interval ordering relationship quantitatively in terms of fuzzy membership for the Game theory. However, the definition of the ordering is in the context for matrix games only. Through recent professional communication with Dr. Dubois, we enhance the initial interval ordering relationship in much rigorous mathematical terms. Prior to the formal definition, we describe the general idea informally first.

Let us start with Figure 3 where '**a** is less than **b**' without any questions. Let us imagine that the interval **a** holds itself and moves right toward **b**. Before the right endpoint of **a** meets the left endpoint of **b**, i.e. $a_R < b_L$, **a** is always less than **b**. At the moment the two endpoints meet, i. e. $a_R = b_L$, all points in **a** are less than all points in **b** except the single common point. Therefore, it is reasonable to believe that **a** is still less than **b**. Let **a** continue its right movement, then **a** and **b** partially overlapped as illustrated in Figure 4. As long $a_L < b_L$ and $a_R < b_R$, except the overlap $[b_L, a_R]$, all points in **a** are less than points in **b**. Also, any point *x* in the overlap $[b_L, a_R]$ is in both **a** and **b**. A point *x* is not less than itself. Therefore, it is reasonable to believe that **a** is still less than **b**.

In the above discussion we believe '**a** is less than **b**' for both cases illustrated in Figures 3 and 4. However, they represent different cases that one has no overlap but the other has. Therefore, we should distinguish them. In fact Fishburn implicitely distinguished them with less than

and weakly less than in [10] for cases illustrated in Figure 3 and 4, respectively. In this paper, we say that **a** is less than **b** in Figure 3 with fuzzy membership one. And, the fuzzy membership of '**a** is less than **b**' illustrated in Figure 4 as 1^- to indicate the fact that **a** is weakly less than **b**. One posible way to describe the weakness is to consider the overlap portion of the intervals.

Now, let us continue to hold **a**, and move it towards the right. With the assumption that the radius of **a** is less than that of **b**, *i.e.* r (**a**) < r (**b**), **a** will be inside **b** completely when and after the left endpoints of **a** and **b** overlaps ($a_L = b_L$) and before the right endpoint of **a** moves outside of **b** ($a_R < b_R$). To keep the continuity, we say that **a** is still less than **b** weakly provided $a_L = b_L$ and $a_R < b_R$. However, when $a_L > b_L$ and $a_R = b_R$, i. e. the right endpoints are aligned and a part of **b** is less than **a**, the membership of '**b** is less than **a**' should be one (actually 1^-) according to our discussion the above. Hence, the statement '**a** is less than **b**' should have the membership zero.

Continuing with the assumption that the radius of **a** is less than that of **b**, we need a membership function for the statement '**a** is less than **b**' that returns one when $a_L = b_L$ and $a_R < b_R$, and zero when $a_L > b_L$ and $a_R = b_R$. We want the membership changes from 1 to 0 continuously. When $\mathbf{a} \subset \mathbf{b}$ and r (**b**) - r (**a**) > 0, the function

$$f(\mathbf{a}, \mathbf{b}) = (b_R - a_R)/2[r(\mathbf{b}) - r(\mathbf{a})]$$
$$= (b_R - a_R)/[(b_R - b_L) - (a_R - a_L)]$$

satisfies the requirements. Hence, we use it to measure the degree of '**a** is less than **b**', when **a** moves from $a_L = b_L$ and $a_R < b_R$, to $a_L > b_L$ and $a_R = b_R$. The fuzzy membership continuously changes from one to zero gradually. Notice the fact that when the midpoints of **a** and **b** overlap, i. e. m (**a**) = m (**b**), then f (**a, b**) returns 0.5.

In the above discussion, we have assumed that r (**a**) < r (**b**). If not, when r (**b**) < r (**a**), by reversing the names of both **a** and **b**, the above arguments valid too. We now consider the case of r (**a**) = r (**b**). All of the above discussions are valid except that the function f (**a, b**) = $(b_R - a_R) / 2[r (\mathbf{b}) – r (\mathbf{a})]$ is undefined since r (**b**) – r (**a**) = 0. However, the function is only used when $a_L = b_L$ and after. In the case of $a_L = b_L$ and r (**a**) = r (**b**), **a** and **b** are actually overlapped completely hence their midpoints as well. Let **c** be an interval whose radius is less than **b**. If we keep the midpoints of **c** and **b** overlap and let the radius of **c** approach that of **a**. The fuzzy membership of '**c** is less than **b**' is 0.5 persistently. Hence, its limit is 0.5 as **c** approach that of **a**. Therefore, the fuzzy membership of '**a** is less than **b**' is 0.5 when r (**a**) = r (**b**) and $a_L = b_L$. It is reasonable that an interval is equally less and greater than itself with fuzzy membership 0.5. Here we need to point out the fact that there is one and only one discontinuity in the above discussion entirely. That appears when r (**a**) = r (**b**) and $a_L = b_L$. The membership has a gap between 1, when r (**a**) < r (**b**) and $a_L = b_L$, and 0.5.

By summarizing the above discussion, we define a fuzzy relationship for two intervals **a** and **b** in terms of fuzzy membership as the follow:

Definition 3: Let $\mathbf{a} = (a_L, a_R)$ and $\mathbf{b} = (b_L, b_R)$ be two intervals, and $\prec$ be a binary interval operator for them. Then, $\mathbf{a} \prec \mathbf{b}$ returns the fuzzy membership for the statement '**a** is less than **b**' as:

1	if $a_R < b_L$
1^-	if $a_L \le b_L \le a_R < b_R$ and r (**a**)> 0
$\frac{(b_R - a_R)}{2[r (b) - r (a)]}$	if $b_L \le a_L < a_R \le b_R$, and r (**b**) > r (**a**)
0.5	if r (**b**) = r (**a**) and $a_L = b_L$

Note: The definition above also works when **a** and **b** are trivial intervals. When both of them are trivial intervals, i.e. $a_L = a_R$ and $b_L = b_R$, the definition returns 1 if $a_R < b_L$, and 0.5 if $a_L = b_L$. It is in consistent with the ordering relation of real numbers. When only one of them is trivial, say **a** is trivial, the definition returns appropriate fuzzy memberships as well.

Definition 3 implies the Corollary below:

Corollary 1: Let **a** and **b** be two intervals. Then

(i) $(\mathbf{a} \prec \mathbf{b}) = 0.5$ iff m (**a**) = m (**b**);

(ii) $(\mathbf{a} \prec \mathbf{b}) > 0.5$ iff m (**a**) < m (**b**);

(iii) $(\mathbf{a} \prec \mathbf{b}) < 0.5$ iff m (**a**) > m (**b**).

Proof:

(i) $\Rightarrow$

Let $(\mathbf{a} \prec \mathbf{b}) = 0.5$. If $\mathbf{a} = \mathbf{b}$, we have m(**a**) = m(**b**). If $\mathbf{a} \neq \mathbf{b}$, then by Definition 3, $b_R - a_R = r (\mathbf{b}) – r (\mathbf{a}) = (b_R - b_L)/2 – (a_R - a_L)/2$. Hence $b_R - a_R = a_L - b_L$. Therefore, $b_L + b_R = a_L + a_R$ and m (**a**) = m (**b**).

$\Leftarrow$

Assume m (**a**) = m (**b**). If $\mathbf{a} = \mathbf{b}$, from Definition 3, $(\mathbf{a} \prec \mathbf{b}) = 0.5$. If $\mathbf{a} \neq \mathbf{b}$, then $b_L + b_R = a_L + a_R$. Hence $b_R - a_R = a_L - b_L = r (\mathbf{b}) – r (\mathbf{a})$. Therefore, $(\mathbf{a} \prec \mathbf{b}) = 0.5$.

(ii) $\Rightarrow$

Let $(\mathbf{a} \prec \mathbf{b}) > 0.5$.

If $(\mathbf{a} \prec \mathbf{b}) = 1$ then $a_R < b_L$ Since m (**a**) $\le a_R$ and $b_L \le$ m (**b**), we have m (**a**) < m (**b**).

If $(\mathbf{a} \prec \mathbf{b}) = 1^-$, then $a_L \le b_L \le a_R < b_R$. Hence, we have $a_L + a_R < b_L + b_R$. This implies m (**a**) < m (**b**).

Otherwise, $(\mathbf{a} \prec \mathbf{b}) = (b_R - a_R) / 2[r (\mathbf{b}) – r (\mathbf{a})] > 0.5$ implies $b_R - a_R > r (\mathbf{b}) – r (\mathbf{a})$, i.e. $b_R - a_R > (b_R - b_L)/2 – (a_R - a_L)/2$. Hence, $b_L + b_R > a_L + a_R$ and m (**a**) < m (**b**).

$\Leftarrow$

Assume m (**a**) < m (**b**). Then, $a_L + a_R < b_L + b_R$ implies $b_R - a_R > r$ (**b**) – r (**a**). If $b_L \le a_L < a_R \le b_R$ and r(**b**) > r (**a**), then (**a** $\prec$ **b**) = $(b_R - a_R) / 2[r$ (**b**) – r (**a**)] > 0.5. Otherwise, (**a** $\prec$ **b**) = 1 or 1^-.

(iii) $\Rightarrow$

Let (**a** $\prec$ **b**) < 0.5. Then, we have $b_R - a_R < r$ (**b**) – r (**a**), i.e. $b_R - a_R < (b_R - b_L)/2 - (a_R - a_L)/2$. Hence, we have $a_L + a_R > b_L + b_R$. This implies m (**a**) > m (**b**).

$\Leftarrow$

Assume m (**a**) > m (**b**). Then, $a_L + a_R > b_L + b_R$ implies $b_R - a_R < r$ (**b**) – r (**a**). Hence (**a** $\prec$ **b**) = $(b_R - a_R) / 2[r$ (**b**) – r (**a**)] < 0.5. □

As the duality of the above discussion, we can define '**a** is greater than **b**' as the follow:

Definition 4: Let **a** = (a_L, a_R) and **b** = (b_L, b_R) be two intervals, and $\succ$ be a binary interval operator that returns the fuzzy membership of the statement '**a** is greater than **b**' as (**a** $\succ$ **b**) = 1 – (**a** $\prec$ **b**).

Similarly, we have the corollary below:

Corollary 2: Let **a** and **b** be two intervals. Then

(i) (**a** $\succ$ **b**) = 0.5 iff m (**a**) = m (**b**);

(ii) (**a** $\succ$ **b**) > 0.5 iff m (**a**) > m (**b**);

(iii) (**a** $\succ$ **b**) < 0.5 iff m (**a**) < m (**b**).

In binary logic, a relation R on a set X is a partial order iff (a) $\forall$ x $\in$ X, xRx $\Rightarrow$ false (inreflexive), and (b) $\forall$ x, y, z $\in$ X, (xRy, yRz) $\Rightarrow$ xRz (transitive) then R is a partial order relation on X.

We now define the concepts of fuzzy inreflexibility and fuzzy transitivity for a fuzzy relation as the follow.

Definition 5: A fuzzy relation R on a set X is fuzzily inreflexive if $\forall$ x $\in$ X, xRx = 0.5; R is fuzzily transitive $\forall$ x, y, z $\in$ X, if xRy > 0.5 and yRz > 0.5 then xRz > 0.5. If R is both fuzzily inreflexive and transitive, then R is a fuzzy partial order relation.

Theorem 1: The binary interval operators $\prec$ and $\succ$ are fuzzy partial order relations.

Proof:

From Definition 3, it is obvious that (**a** $\prec$ **a**) = 0.5 since r (**a**) = r (**a**) and $a_L = a_L$. Therefore, the binary operator $\prec$ is fuzzily inreflexive.

Let **a**, **b** and **c** be nontrivial intervals. From Corollary 1 we have (**a** $\prec$ **b**) > 0.5 $\Rightarrow$ m (**a**) < m (**b**) and (**b** $\prec$ **c**) > 0.5 $\Rightarrow$ m (**b**) < m (**c**). The midpoints of intervals are just real numbers. Hence (**a** $\prec$ **b**) > 0.5 and (**b** $\prec$ **c**) > 0.5 imply m (**a**) < m (**c**). Therefore, the binary operator $\prec$ is fuzzily transitive. Hence, it is a fuzzy partial order.

Similarly, the binary interval operator $\succ$ is a fuzzy partial order. □

We have now established fuzzy partial orders for intervals. Definitions 3 and 4 provide quantitative methods to find the exact fuzzy memberships for ordering two intervals. From Corollaries 1 and 2, we can see that if the fuzzy membership is greater (or less) than or equal to 0.5 is in fact determined by their midpoints only.

We use a few examples to complete this section:

For the two nested intervals **a** = [0, 4] and **b** = [1, 3], the fuzzy memberships for '**a** is less than **b**' and '**a** is greater than **b**' are both 0.5 since their midpoints overlap.

Let **a** = [0, 4] and **b** = [2, 4], then '**a** is less than **b**' has a fuzzy membership of one minus while the fuzzy membership for '**a** is greater than **b**' is zero.

For the intervals **b** = [0, 5] and **a** = [1, 3], '**a** is less than **b**' has a fuzzy membership of 2/3 while the fuzzy membership of '**a** is greater than **b**' is 1/3.

4. CRISP AND FUZZY SHORTEST PATHS FOR INTERVAL WEIGHTED GRAPHS

In this section, we study shortest paths for interval weighted graphs by applying the fuzzy partial order relations for intervals defined in the previous section. Among the algorithms of finding shortest paths, Dijkstra's algorithm [8] (1959) is probably the most well-known. It finds the shortest paths from one vertex to the rests in a connected, undirected graph with a growing 'cloud'. In this section, we extend Dijkstra's algorithm for interval weighted graphs first and then others. For the readers' convenience, we provide the pseudocode for Dijkstra's algorithm from [13] below.

Algorithm: DijkstraShortestPath (G, v)

Input: *A simple undirected graph G with nonnegative edge weights, and a vertex v of G.*

Output: *A label D[u] for each vertex u of G, such that D[u] is the distance from v to u in G.*

```
for all u∈G.vertices()
    if u = v, D[u] ← 0
    else D[u] ← ∞
Let a priority queue Q contain all
  vertices of G using the D labels as keys
while ¬Q.isEmpty()
    u←Q.removeMin()
∀ vertex z adjacent to u and z ∈ Q do
 {perform relaxation on the edge(u, z)}
   if D[u] + w(u, z) < D[z] then
       D[z] ←D[u] + w(u, z)
      {change the key of vertex z in Q
         with D[z]}
return the label D[u] of each vertex of G
```

As we can see in the Dijkstra's algorithm, the most critical step is to compare the distance labels of *D[u] + w (u, z)* and *D[z]* after their initialization. Whenever D[u] + w (u, z) < D[z], we update D[z] by D[u] + w (u, z) in the priority **Q**. This is called edge relaxation. Only the edge with minimum distance is added in to form a shortest path consequently.

With the partial order relationship defined in the previous section for intervals, we can modify the above Dijkstra's algorithm for interval weighted graphs with interval

comparisons. The only modification that needs to be made for the interval Dijkstra's algorithm is that in each step of the edge relaxations, we need to find the fuzzy membership for the label `D[z]` being the least. We then select the interval edge, which makes `D[z]` being the least with the maximum membership, into Dijkstra's 'cloud'. The maximum can be obtained with the 'drastic sum' operation or others in fuzzy logic [23] and [24]. Let us make the following comments to distinguish two kinds of shortest paths in an interval weighted graph:

- (**Strong crisp and crisp shortest paths**) If all edges of a path brought into Dijkstra's 'cloud' have fuzzy memberships one or one minus to be the least weight, then we say that the path is the shortest crisp (or strongly crisp if in fact the memberships are all one but minus). For example, the shortest paths of the interval weighted graph G in Figure 2 from A to C, E, F, are strongly crisp. The shortest path from $A \rightarrow C$ has an interval weight [2, 4]. The shortest path from $A \rightarrow C \rightarrow E$ has an interval weight [3, 7]; and $A \rightarrow C \rightarrow E \rightarrow F$ has an interval weight [6, 12].
- **(Fuzzy shortest paths)** Otherwise, shortest paths found are fuzzy. During the generation of a fuzzy shortest path, a vertex is introduced through the edge relaxation. Edge relaxations are done through interval comparisons. Hence, a sequence of fuzzy memberships comes along with the construction of the path.

We now can define the fuzzy membership of a fuzzy shortest path as the follow:

Definition 6: Let P be a fuzzy shortest path from a vertex A to B produced with the interval Dijkstra's algorithm and μ_e be the fuzzy membership of an edge $e \in P$ at the time it brought in the 'cloud' of being the least D[z] for all other edges in E\P. Then the fuzzy membership of the fuzzy shortest path P is $\nu_P = \min \mu_e$ for all $e \in P$.

The above definition comes from the 'drastic product' in fuzzy logic [23] and [24]. Of course one may use bounded difference, Einstein product, Hamacher product, or others to define the fuzziness of a fuzzy shortest path depending on ones preference. However, if there is a shortest path from A to B with membership greater than 0.5 in an interval weighted graph then it is the only fuzzy shortest path from A to B with membership greater than 0.5. We state it as a theorem below with a proof.

Theorem 2: In an interval weighted graph, if P is a fuzzy shortest path from A to B with membership greater than 0.5 then it is the only path from A to B with membership greater than 0.5.

Proof:

Let P and P' be two distinct shortest paths from A to B and both have membership greater than 0.5 as illustrated in Figure 6. Without loss of generality, we assume CD and CK are the first different edges between P and P'. As we defined, the fuzzy membership of the fuzzy shortest path is the least element in E\P. So when the algorithm choose CD as an edge of P it must has compared with CK, that means $D[D] \prec D[K]$ is greater than 0.5. Hence, $D[K] \prec D[D]$ is less than 0.5 according to Definition 4. Therefore, the membership of P' being a shortest path is less than 0.5. This is a contradiction. Hence P and P' are identical.

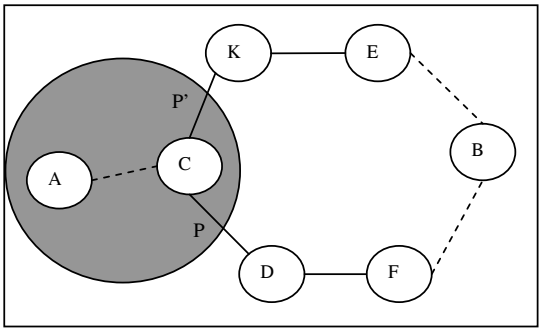

Figure 6: The uniqueness of a shortest path with

Theorem 2 implies the uniqueness of interval shortest path with fuzzy membership greater than 0.5. However, it should be pointed out that if the fuzzy membership of interval shortest path is close to 0.5 then it may not be helpful in practice at all. Hence, one may want to use a predetermined level $\alpha \in [0.5, 1]$ for the minimum allowable fuzzy membership for to be generated shortest paths. If one sets the α value too high in applications, then the interval Dijkstra's algorithm may generate no shortest paths at all.

We use an example in Figure 7 to summarize the above discussion.

Example: As illustrated in Figure 7 below, we apply the extended interval Dijkstra's algorithm with the drastic sum in the literature of operations of fuzzy sets,. The shortest path from A to F is AC – CB – BE – EF with a fuzzy membership 0.8.

If we use $\alpha > 0.9$ then the interval Dijkstra's algorithm generates no shortest paths from A to F.

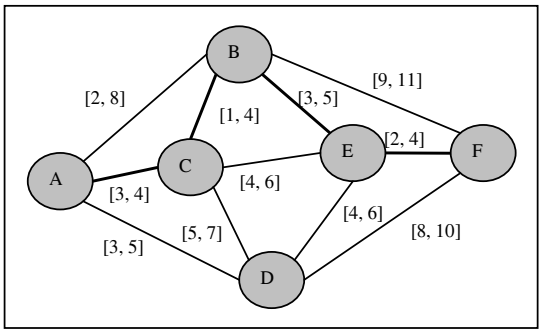

Figure 7: Fuzzy shortest path in an interval weighted graph

As we know the partial order relationship 'less than' of real numbers plays the fundamental role in all currently available shortest path algorithms. Therefore, by using the interval fuzzy partial order relationship we defined, we may extend other existing shortest path algorithms as well.

In the classical Dijkstra's algorithm it is assumed that there are no negative-weight edges in a graph. By assuming edges are directed and no negative-weight cycles, the Bellman-Ford algorithm [2] finds shortest paths of a graph even with negative weighted edges. The pseudo code below is from [13]

```
Algorithm BellmanFord(G, s)
    for all  v ∈ G.vertices()
        if  v = s
            setDistance(v, 0)
        else
            setDistance(v, ∞)
    for i ← 1 to n-1 do
    for each  e ∈ G.edges()
        {relax edge e}
        u ← G.origin(e)
        z ← G.opposite(u,e)
        r ← getDistance(u)+ weight(e)
        if  r < getDistance(z)
            setDistance(z,r)
```

If we use the interval partial order for the statement `if r < getDistance(z)` in the algorithm above, we can apply it directly for interval weighted graphs. Similarly these extended algorithms may produce crispy and/or fuzzy shortest paths. We can similarly extend other shortest path algorithms for interval weighted digraphs as well.

5. MINIMUM SPANNING TREE FOR INTERVAL WEIGHTED GRAPHS

In connected graph G = (V, E), one may remove some edges to form a sub-graph G' = (V, E') with the same number of vertices but less number of edges. If G' is in fact a tree T, then it is called a spanning tree of G. If G is weighted, then among all of its spanning trees the one with the minimum total weight is called the minimum spanning tree (MST) of G.

Algorithm: KruskalMST (G)

```
for each vertex v in G do
      define a Cloud(v) of  {v}
let Q be a priority queue.
  Insert all edges into Q using their
  weights as the key
  T ← ∅
while T has fewer than n-1 edges do
          edge e = Q.removeMin()
  Let u, v be the endpoints of e
  if Cloud(v) ≠ Cloud(u) then
      Add edge e to T
      Merge Cloud(v) and Cloud(u)
  return T
```

The first algorithm that finds a minimum spanning tree was developed by Czech scientist Borůvka in 1926 [5]. Its purpose was to find efficient electrical coverage of Bohemia. Other algorithms to find MST include Kruskal's MST algorithm [16] (1956) and Prim-Jarnik's MST algorithm [19] (1957). Most MST algorithms, if not all, take the greedy approach. Hence, sorting is often required according to given partial ordering relations. The partial order relations we developed in section 3 of this paper can be applied to extend these algorithms to find minimum spanning trees for interval weighted graphs. As an example, we extend the Kruskal's MST algorithm for interval weighted graphs. For reader's convenience, we copy the Kruskal's MST algorithm in pseudo code from [13] as the above.

In the above algorithm the most critical step is to construct a priority queue, Q, using the weights of edges as the key. The fuzzy binary interval operator $\prec$ is a fuzzy partial order for intervals as we proved in Theorem 1. Therefore, to extend the MST algorithm, the only thing we need to do is to apply the fuzzy partial order relation to form a priority queue according to the interval weights associated with the edges.

Let us use a heap to implement the priority queue. Then, the interval weighted edges are stored in a balanced binary tree. We denote the interval weighted edge stored in the node m of the binary tree as e_m. Let *i* be a node of the binary tree, and j and k be its immediate children. Then, to ensure the heap condition both $e_i \prec e_j$ and $e_i \prec e_k$ should be at least 0.5. Hence, the interval in the root r is the least interval in terms of the fuzzy partial order in Definition 3. Let us use m and n to denote the two immediate children of r. Then, the fuzzy membership for e_r being the least interval in the heap can be reasonably defined as $\sigma_r = \min \{e_r \prec e_m, e_r \prec e_n\}$ which is no less than 0.5.

With the heap implementation for the extended Kruskal's algorithm, the number of down heaps for each re-heap is at most log M, where M is the number of edges of the interval weighted graph. Hence, the overall asymptotic complexity of the extended Kruskal's algorithm is still O (M log M).

We now discuss the fuzzy membership of the generated MST by removing interval weighted edges from the root of the heap consecutively. There should be total |V| -1 interval edges de-queued from the root such that each of them has its own fuzzy membership of the least interval in the heap before the de-queue operation. Of course, if for every de-queued interval edge its fuzzy membership of being the least intervals is one or one minus then the interval MST is crisp. Otherwise, let μ_T denote the fuzzy membership of the minimum spanning tree. One idea is to define $\mu_T = \min \sigma_r$. However, it can be misleading. Assume there are two interval weights that are equal or have the same midpoints and both of them are 'short' enough to be able to make the root in the heap and to be included in the MST. The fuzzy membership of 0.5 for the tree would not make any sense. Another approach is to define $\mu_T = \max \sigma_r$. However, this maybe misleading as well since any single crisp comparison will result in a crisp MST.

Both $\mu_T = \min \sigma_r$ and $\mu_T = \max \sigma_r$ that are most commonly used in fuzzy logic can be misleading so we need reasonable alternatives. In applications, one may want to output the fuzzy memberships as well when perform the de-queue operations. By observing their behavior, one may have a better sense of the fuzziness of the generated MST. Another approach is to use the fuzzy membership of the last edge de-queued from the heap

when generating an MST by interval extended Kruskal's algorithm.

Similarly, we can extend other available MST algorithms for interval weighted graphs. We finish this section with an example of interval MST.

Example: As G in Figure 7, we find a minimum spanning tree of G with extended interval Kruskal's algorithm. The MST is formed by AC, CB, BE, ED, EF with its total interval weight is [13, 24].

6. Conclusion and future work

In this paper, we have introduced interval-valued weights to model the uncertainty and variability in weighted graphs. In order to study application algorithms for interval weighted graphs, we have defined continuous (except at a single point for two equal intervals) binary interval operators that can quantitatively describe if one interval is less than or greater than another. We proved that the operators in fact form fuzzy partial order relationship for intervals. By applying the partial order relationship, we have extended application algorithms that find shortest paths and minimum spanning tree for interval weighted graphs. These can be directly applied to handle interval valued uncertainties in decision making systems modeled by weighted graphs.

Our work here is just an initial investigation on interval weighted graphs. We have searched the internet extensively for 'interval weighted graph' in the literature but not very successfully. The relationship of intervals was first studied by Allen [1] (1983). His study was mainly for temporal properties described as 13 cases. Another close match was Fishburn's work on interval orders and interval graphs [10] in 1985. However, the purpose of that study was mainly for ordering vertices in a graph but not for interval weighted edges at all.

The theoretic results on interval fuzzy partial order relations can have more potential applications in dealing with systems with interval valued uncertainties other than interval weighted graphs. We are working on other applications, and report our results on scheduling tasks on flow networks with temporal uncertainty in [14].

Acknowledgement: This research is partially supported by NSF Grant CCF-0202042.

Authors of this paper appreciate the insight comments from Dr. D. Dubois on our previous work. By studying his comments we have been able to present the fuzzy partial order relationship for intervals in much clearer terms in this paper.

REFERENCES

[1] Allen, J. F. *Maintaining Knowledge about Temporal Intervals*, Communication of the ACM, Vol. 26, pp.832-843, 1983.

[2] Bellman, R. *On a Routing Problem*, in Quarterly of Applied Mathematics, 16(1), pp.87-90, 1958.

[3] Benzer, S. *On the topology of the genetic fine structure*. Proc. Nat. Acad. Sci. USA, 45, pp.1607-1620, 1959.

[4] Booth, K. S. and Lueker, G. S. *Testing the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms*, J. Comput. Syst. Sci., 13, pp. 335-379, 1976.

[5] Borůvka, O. *On a certain minimal problem,*1926

[6] Collins, D. and Hu, C. *Fuzzily Determined Interval Matrix Games,* Forging New Frontiers: Fuzzy Pioneers II, edited by M. Nikravesh, J. Kacprzyk and L. Zadeh, in press, 2006.

[7] Cormen, T. H., Leiserson, C. E., Rivest, R. L. and Stein, C. *Introduction to Algorithms*, Second Edition. MIT Press and McGraw-Hill, Section 23.2: pp.567–574, 2001.

[8] Dijkstra, E. W. *A note on two problems in connexion with graphs*. Numerische Mathematik, Vol. 1, pp. 269-271, 1959

[9] Ford, L. R., Fulkerson, D. R. *Flows in Networks*, Princeton University Press, 1962.

[10] Fishburn, P. C. *Interval Orders and Interval Graphs: A study of Partially Ordered Sets*, Wiley, New York, 1985.

[11] Gilmore, P. C. and Hoffman, A. J. *A characterization of comparibility graphs and interval graphs*, Canadian Journal of Mathematics, 16, pp.539-548, 1964.

[12] Golumbic, M. C. *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, 1980.

[13] Goodrich, M. and Tamassia, R. *Algoritm Design*, John Wiley & Sons, 2002.

[14] Hu, P., Deallar M., and Hu, C. Task scheduling on flow networks with temporal uncertainty, Proc. of IEEE 2007 Symposium on Foundations of Computational Intelligence, Honolulu, HI, 2007.

[15] Interval Computations, http://www.cs.utep.edu/interval-comp/main.html

[16] Kruskal, J. B. *On the shortest spanning subtree and the traveling salesman problem*, Proceedings of the American Mathematical Society 7, pp.48–50, 1956.

[17] Moore, R. E. *Method and Application of Interval Analysis*, SIAM, Philadelphia.

[18] Nguyen, H., Kreinovich, V. and Longpre, L. *Dirty Pages of Logarithm Tables, Lifetime of the Universe, and (Subjective) Probabilities on Finite and Infinite Intervals*, Reliable Computing, 10, No. 2, pp. 83-106, 2004.

[19] Prim, R. C. *Shortest connection networks and some generalizations*, *Bell System Technical Journal*, 36, pp. 1389–1401, 1957.

[20] Sengupta, J. *Optimal Decision Under Uncertainty*, Springer. New York. 1981.

[21] Wagman, D., Schneider, M., Shnaider, E. *On the use of interval mathematics in fuzzy expert systems*, International Journal of Intelligent Systems, 9, pp. 241-259, 1994.

[22] West, D. B. *Introduction to graph theory*, Prentice Hall, Inc., Upper Saddle River, NJ, 1996.

[23] Yan, J. and Langari, R. *Fuzzy Logic: Intelligence, Control and Information*, Prentice-Hall, 1999.

[24] Zadeh, L. *Fuzzy sets*, Information Control, 8, pp. 338-353, 1965.