Fuzzy Temporal Graphs and Sequence Modelling in Scheduling Problem

Margarita Knyazeva\textsuperscript{1(✉)}, Alexander Bozhenyuk\textsuperscript{1(✉)}, and Uzay Kaymak\textsuperscript{2(✉)}

\textsuperscript{1} Southern Federal University, Taganrog, Russian Federation
mkyazeva@sfedu.ru, avb002@yandex.ru
\textsuperscript{2} School of Industrial Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands
u.kaymak@ieee.org

Abstract. Processing sequential data and time-dependent data is a problem of constructing computational graph with a certain structure. A computational graph formalizes the structure of a set of computations including mapping temporal inputs and outputs. In this paper we apply graph theory and fuzzy interval representation of uncertain variables to indicate states of the temporal scheduling system. Descriptive model for temporal reasoning on graph, sequence modelling and ordering of fuzzy inputs for scheduling problem is introduced.

Keywords: Fuzzy sequence modelling \cdot Computational graph \cdot Fuzzy graph \cdot Fuzzy temporal intervals \cdot Temporal reasoning \cdot State-transition system

1 Introduction

Temporal reasoning and temporal knowledge representation are the problems of introducing relations between elements or events in time with respect to known information and precedence relations between those events. Fuzzy temporal reasoning can be divided in two main approaches, depending on how time is represented: quantitative and qualitative. The quantitative approach is relevant when temporal data and temporal stamps are available and extracting new knowledge is necessary, while qualitative approach investigates relative fuzzy relations between elements or events - such as event A happens before event B, event C happens during event B and one needs to produce inferences on the known temporal facts.

Within both approaches there are two main problems to be solved: how to represent basic fuzzy units of time and handle it (how to “quantify time and measure it”) and how to represent fuzzy relationships between basic units of time (how to “sequence events and locate them”). A number of different approaches exist to handle both of these problems: topological ordering techniques and algorithms, Allen’s crisp interval algebra \cite{1, 2}, fuzzy intervals \cite{3} and probabilistic intervals, computational graphs, state-variable representation \cite{4} and sequence modelling techniques. These approaches are at the basis of proposed extensions to knowledge representation on, for example, the world wide web (see, e.g. \cite{5}).
Measurement of time means the measurement of the uncertain temporal duration of events or “start-start” or “start-finish” fuzzy intervals, while specification means locating these events within the timeline and sequencing them. Locating an event in the time series means in the first instance locating it relative to other events.

2 Fuzzy Sequence Modelling and Temporal Interval Knowledge Representation

Usually we refer to sequence modelling when we use recurrent neural networks, or RNNs as a family of neural networks for processing sequential data. Sequence modelling in scheduling is the problem of predicting what event comes next: the current output is dependent on the previous input or state. In this paper we suggest temporal knowledge representation and sequence modelling technique for scheduling on a fuzzy graph.

The most traditional framework for handling the qualitative relations between time-dependent intervals is Allen’s Interval Algebra, or Interval Algebra (IA), formalized for the first time in [2]. The basic idea of IA is modelling temporal events as intervals, that have a starting and a finishing points in time. Based on this idea, 13 different temporal relations may be introduced between any given pair of events.

Table 1. Precedence relations between events according to Allen’s Interval Algebra.

| Basic precedence relations | Graphical illustration | Formal notation |
|---------------------------|------------------------|-----------------|
| x before y y after x     | xxx < yyyy             | x− < y         |
| x meets y y is met by x  | m xxxx m~ yyyy         | x+ = y−        |
| x overlaps y y is overlapped by x | o xxxx o~ yyyy | x− < y− < x+ < y+ |
| x during y y includes x  | d xxxx d~ yyyy yyyy   | x− > y         |
| x starts y y is started by x | s xxx s~ yyyy yyyy yyyy | x− = y         |
| x finish y y is finished by x | f xxx f~ yyyy yyyy yyyy | x− > y−        |
| x equals y               | ≡ xxxx yyyy            | x− = y         |

Table 1 shows the 13 relations between events performance, formalizing them with logical constraints (x− denotes the left end of the time interval x and x+ denotes the right end of x).

Temporal ordering of events may be expressed in terms of precedence relations and can be represented in the form of a graph, forming a type of graph that is called a temporal interval graph. However, in the real world, the actual start and finish times of
many events may be hard to define, and so, in the fuzzy temporal interval graph, relations between events are defined in terms of degrees of truth, allowing multiple relations between elements at the same time. Figure 1 illustrates this kind of fuzzy graph relations between events.

![Fuzzy Temporal Graph Diagram](image)

**Fig. 1.** Qualitative fuzzy relations between events in graph.

The truth value of a relation is the degree of “existence” of that relation, varying from 0 to 1. For example, in Fig. 1, Event-2 and Event-4 are related with propositions \( \text{precedes}(\text{Event-2, Event-4, 0.7}) \), or relations like \( \text{meets}(\text{Event-2, Event-3, 0.6}) \). Fuzzy edge between any two events is defined by a tuple that may be composed of multiple truth values for different fuzzy precedence relations.

The quantitative analysis for fuzzy temporal data can be done with the help of fuzzy interval representation of the areas near the edge of the interval that have less membership in the time-unit than the ones near the center, or simply fuzzy membership function. Gradual numbers as elements of fuzzy intervals were introduced as tools for computations on fuzzy sets, usually associated with combinatorial optimization and monotonic function evaluation.

**Definition 1.** Let \( X \) denote a set, then a fuzzy set \( \tilde{A} \) on \( X \) is a set of ordered pairs \( \tilde{A} = \{(x, \mu_{\tilde{A}}(x) : x \in X)\} \), where the membership function \( \mu_{\tilde{A}}(x) : X \rightarrow [0, 1] \) is a map from set \( X \) into the set of possible degrees of memberships with \( \mu_{\tilde{A}}(x) = 1 \) indicating full membership, \( \mu_{\tilde{A}}(x) = 0 \) indicating non-membership.

The \( \alpha \)-cut of a fuzzy set is the set \( \{x | \mu_{\tilde{A}}(x) \geq \alpha \} \) and is denoted as \( \tilde{A}_{\alpha} \).

**Definition 2.** A fuzzy interval \( M \), defined by its membership function \( \mu_M(\cdot) \), is a fuzzy subset of the real line such that, if \( \forall (x, y, z) \in \mathbb{R}^3, z \in [x, y], \) then \( \mu_M(z) \geq \min(\mu_M(x), \mu_M(y)) \) [6].
A fuzzy interval $M$ is normal iff $\exists x \in \mathbb{R}$ such that $\mu_M(x) = 1$. A fuzzy interval is a normalized fuzzy set whose membership function is upper-semicontinuous and whose $\alpha$-levels are convex.

**Definition 3.** The set $\{x | \mu_M(x) = 1\}$ is the core of a fuzzy interval (Fig. 2). As fuzzy intervals are assumed to be normal, fuzzy intervals with membership functions and the $\alpha$-cut of a fuzzy interval can be stated as follows: $M_\alpha = \{x | \mu_M(x) \geq \alpha \geq 0\}$.

Let $M_1 = \{x | \mu_M(x) = 1\}$ and $M_1 = [m_1^-, m_1^+]$ is the core of a fuzzy interval, then its support is an open interval:

$$M_0 = \text{closure}\{x | \mu_M(x) > 0\} = \text{closure}\{(m_0^-, m_0^+)\} = [m_0^-, m_0^+]$$ and $\mu_M$ is non-decreasing on $(-\infty, m_1^+]$ and $\mu_M$ is nonincreasing on $[m_1^+, +\infty)$.

![Interval L-R representation for earliest and latest activity starting dates for a scheduling problem.](image)

For practical purposes and ease of their use fuzzy intervals can be presented as left and right parts, or L-R representation of fuzzy interval [7]. Fuzzy interval can be defined by its membership function $\mu_M$, core $[m^-, m^+]$, a support $[m^- - \varepsilon_M, m^+ + \beta_M]$ as well as L and R reference functions as follows:

$$\mu_M(x) = \begin{cases} 1 & \text{for } x \in [m^-, m^+], \\ L\left(\frac{m^- - x}{\varepsilon_M}\right) & \text{for } x < m^- , \\ R\left(\frac{x - m^+}{\beta_M}\right) & \text{for } x > m^+. \end{cases} \tag{1}$$

In the literature of fuzzy optimization, the idea of fuzzy solution was introduced by Verdegay [9], who pays special attention to duality among fuzzy constraints and fuzzy objectives in fuzzy linear mathematical programming problems. Recently several
investigations were done by Dubois, Fargier and others \[8, 10, 11\] about the notion of fuzzy elements and interval analysis for optimization methods. In this paper we suggest using fuzzy interval representation and gradual numbers for scheduling problems to deal with uncertainty in operational planning. Fuzzy interval here is the interval that reflects incomplete knowledge about some parameter that lies between two bounds, so that a value within the interval is possible and a value outside is impossible. Making uncertainty gradual means making the boundaries of the interval softer, and so fuzzy boundaries of fuzzy intervals should be gradual \[6\].

**Definition 4.** A gradual number \(\tilde{r}\), is defined by a function, called the “assignment function” \(A_{\tilde{r}} : (0, 1) \to \mathbb{R}\). Then for each \(\alpha\), a real value \(r_\alpha\) is given by \(A_{\tilde{r}}(\alpha)\). A gradual real number can be understood as a real value parametrized by \(\alpha\).

Let \(\mu_M^+\) and \(\mu_M^-\) be the parts of the membership function of a fuzzy interval \(M\). They are functions from the real line to \([0, 1]\), respectively, defined on \([m_0, m_1]\) and \([m_0^+, m_1^+]\). These functions are injective (\(\mu_M^+\) is increasing and \(\mu_M^-\) is decreasing) and \((\mu_M^-)^{-1}(x)\) and \((\mu_M^+)^{-1}(x)\) are their inverse functions and the endpoints of the \(\alpha\)-cut of \(M\). The domain of a gradual number is defined as \((0, 1]\) so that it represents all the possibility degrees for which the \(\alpha\)-cuts are defined. Fortin, Dubois and Fargier treated fuzzy interval as a set of gradual numbers that lie between two gradual number end-points in the same way that a real interval can be treated as a set of real numbers that lie between two real end points \[6\].

In this paper we suggest interpreting fuzzy interval \(M\) for earliest and latest starts of events as an interval with the \(\alpha\)-cut mapping \(\alpha \to M_\alpha\) as an assignment function from \((0, 1]\) to the set of intervals.

Usually most algebraic properties of real numbers are preserved for gradual numbers, comparing with fuzzy intervals, but gradual real numbers are not totally ordered \[6\]. Using gradual numbers, we can introduce a fuzzy interval \(M\) for event performance by an ordered pair of gradual numbers \((\tilde{m}^-, \tilde{m}^+)\), where \(\tilde{m}^-\) is called left profile or earliest starting time of event, while \(\tilde{m}^+\) is the right profile or latest starting time (so-called lower and upper bounds). Such profiles of fuzzy sets are piecewise linear and can be easily implemented while estimating uncertain interval values, where a fuzzy bound is a gradual number. Algebraic operations and algebraic status of gradual numbers were discussed in several works \[11–14\] concerning graph-based scheduling problems. Other applications of gradual numbers can be implemented in path of maximal capacity problem, shortest path problem, allocation problems and other optimization problems, which can be formalized as linear programming problems.

### 3 Interval Temporal Modelling and Project Scheduling Problem

Usually a scheduling problem is characterized by precedence relations – an activity cannot start before its preceding activities are not finished; temporal constraints – an activity \(i\) cannot start before its earliest start \(e_i\) and \(b_i\) is the date after which activity cannot be started without delaying the end of the project (latest start time); capacity constraints – each activity \(i\) requires a certain level of resources for its execution at each
moment $t$, those resources are limited and can be renewable and non-renewable. The critical path method (CPM) is an algorithm for scheduling a set of project activities, based on evaluation of temporal constraints. A critical path is determined by identifying the longest path of dependent precedence-related activities and measuring the time required to complete them from starting activity to finishing one.

Several main temporal variables need to be calculated based on temporal fuzzy intervals for CPM, gradual formalization and precedence relations for scheduling problem:

**The earliest starting date** $e_i$ of an event $i$ or activity is the date before which it can’t be started without violation of a precedence constraint.

**The latest starting date** $b_i$ is the date after which we cannot start the activity without delaying the end of the project.

**The float** is the difference between the latest starting date and the earliest starting date of the activity calculated as follows: $f_i = b_i - e_i$. An activity is considered to be critical iff its float is null.

According to Critical Path Method approach to resource-constrained scheduling problem we use a *forward graph propagation* to determine the earliest starting and finishing dates (finally project duration and the free floats) and a *backward graph propagation* to determine the latest starting and finishing dates.

$P_{ij}$ denotes the set of all paths from activity $i$ to activity $j$, while $T(p_{ij})$ is the temporal length of the path $p_{ij} \in P_{ij}$ in graph, while the length of longest path is the earliest starting date $e_i$ from starting activity 1 (sink) to activity $i$ (target) so that the following condition is true:

$$e_i = \max\{T(p_{ij}) | p_{ij} \in P_{ij}\}.$$ (2)

The latest starting date is calculated as follows:

$$b_i = \max\{T(p_{ij}) | p_{ij} \in P_{1,n}\} - \max\{T(p_{ij}) | p_{ij} \in P_{i,n}\}.$$ (3)

CPM algorithm is based on the estimation of the longest possible continuous path in the graph, taken from the initial event to the terminal event so that:

$$e_i = \max\{e_j + d_j | j \in Pred_i\}$$ (4)$$b_i = \min\{b_j - d_j | j \in Succ_i\},$$ (5)

where $d_j$ is the activity duration, $Pred_i$ and $Succ_i$ are the immediate predecessors and successors of the activity $i$. For each activity $i$ in the graph, the three functions depending on the number of activities $n$ are to be introduced: $e_i(\cdot)$, $b_i(\cdot)$ and $f_i(\cdot)$. 
For example, earliest starting time calculated by forward graph propagation is increasing according to each argument:

$$\tilde{e}_i^{-} = \max\{\tilde{e}_i^{-} + \tilde{d}_j^{-} | j \in \text{Pred}_i\}$$  \hspace{1cm} (6)

$$\tilde{e}_i^{+} = \max\{\tilde{e}_i^{+} + \tilde{d}_j^{+} | j \in \text{Pred}_i\}.$$  \hspace{1cm} (7)

Figure 3 illustrates fuzzy interval representation of temporal variable for earliest and latest starting times of activity $i$.

However the recursive backward graph propagation approach to determine the latest starting and finishing dates $b_i$ is not appropriate due to the nature of the uncertainty that would be considered twice and the necessity to rank fuzzy numbers constantly due to the technique itself. Finally fuzzy subtraction increases the imprecision in the sense that the result of $\tilde{b}_i - \tilde{d}_i$ is more imprecise than $\tilde{b}_i$ or $\tilde{d}_i$ separately. Therefore variables $\tilde{b}_i$ and $\tilde{d}_i$ are non-interactive. Dubois and Prade introduced the concept of weakly non-interactive fuzzy numbers whose operations are based on the extension principle corresponding to each t-norm in place of the minimum operator [15]. Based on the idea of fuzzy interval calculations of floats and latest times, Zielinski introduced polynomial algorithms for determining the intervals of the latest starting times in scheduling networks and complexity results for floats [13].

Thus the fuzzy interval containing the float of activity $i$ cannot be calculated by subtracting the fuzzy earliest starting time from the fuzzy latest starting time.

In this paper we consider arithmetic on intervals of gradual numbers [17], which uses the arithmetic on functions of the L-R profiles as a generalization of the real-valued interval analysis.
4 State-Transition and Time-Oriented Modelling

A project scheduling problem (PSP) can be defined by a number of events or activities representing different planning states of a project (or partial schedules) and a set of precedence relations between events. The problem can be presented as an event-network or a directed acyclic graph where vertices represent events while arcs illustrate precedence constraints. The idea of a project manager is generally to minimize the makespan of the project with respect to resources.

To perform an event, different kinds of resources may need to be assigned and consumed. Time is a resource required by every action or event to be performed, but it differs from other types of resources. We assume that every event performance produces an immediate transition from one state to another in the network, depending on the previous states, so temporal models must specify every state at various points during the project. General linear programming formulation of resource-constrained fuzzy scheduling problems is given in [16].

Temporal modelling and scheduling usually presuppose constraints and deadlines, events that may be expected to occur at future time periods and to be located in time with respect to deadlines and resource usage. An actor usually needs predictive model of its actions to decide what actions to do and how to do them. Thus, two types of models can be introduced: descriptive and operational models, respectively.

**Descriptive models** of actions or events describe which state or set of possible states may result from performing an action or event. **Operational models** describe how to perform an action, to carry out an activity, what operation to execute in the current context, and how to organize them to optimize the objective function using interval graph representation. Fuzzy interval-valued graph or fuzzy intersection graph is a fuzzy graph showing intersecting intervals on the real line so that each vertex is assigned an interval and two vertices are joined by an edge if and only if their corresponding intervals overlap.

**Definition 6.** By an interval-valued directed fuzzy graph of a graph \( \tilde{G} = (V, E) \) we mean a pair \( \tilde{G} = (A, R) \), where \( A = [\mu_A^-, \mu_A^+] \) is a left-right interval-valued fuzzy set on the set of vertices \( V \) and \( R = [\mu_R^-, \mu_R^+] \) is an interval-valued fuzzy relation on \( E \).

**Definition 7.** A fuzzy interval \( M \) is defined by an ordered pair of gradual numbers \( (\tilde{m}^-, \tilde{m}^+) \), where \( \tilde{m}^- \) is called the fuzzy lower bound or left-profile and \( \tilde{m}^+ \) is called the fuzzy upper bound or right-profile, and \( A_M \) is an assignment function.

**Property 1.** The domains of the assignment functions \( A_{\tilde{m}^-} \) and \( A_{\tilde{m}^+} \) must be \((0, 1]\).

**Property 2.** Assignment function \( A_{\tilde{m}^-} \) must be non-decreasing and assignment function \( A_{\tilde{m}^+} \) must be non-increasing.

**Property 3.** Lower and upper bounds \( \tilde{m}^- \) and \( \tilde{m}^+ \) must be such that \( A_{\tilde{m}^-} \leq A_{\tilde{m}^+} \) for all \( \alpha \)-cuts.

The precise organization of a hierarchy of data structures and state representations is a well-known area in computer science. Scheduling problems presuppose some decisions about when and how to perform a given set of actions with respect to time constraints, resource constraints and the objective function. They are typically NP-complete.
Definition 8. A State-transition system or planning domain is a 4-tuple $\Sigma = (S, A, \gamma, \text{cost})$ where:

- $S$ – is a finite set of states for the system,
- $A$ – is a finite set of actions to perform,
- $\gamma$ – is a state-transition function, that gives the next state, or possible next states, after an action or event, $S \times A \rightarrow S$ with $\gamma(s, a)$ being a predicted outcome.
- Cost – is a partial function, $S \times A \rightarrow [0, \infty)$ having the same domain as $\gamma$. The cost function may represent monetary cost, time, or other resources to minimize.

Each state of a system represents a time-oriented partial plan (schedule or interval-valued graph) that is associated with a certain timeline. Figure 4 illustrates two-dimensional state-time oriented model for a scheduling problem. State-oriented partial schedule keeps the notion of the global states and transition between the states (complete descriptions of the domain at some time point), while time-oriented profile represents the dynamics as a collection of partial intervals or primitives in time.

![Fig. 4. Interval two-dimensional state-time oriented model.](image)

Definition 9. The planning computational graph is a graph that consists of a number of states corresponding to timeline and computations or number of actions made according to the plan. Each state is associated with a number of inputs and outcome. Planning graph is an approximation of a complete enumeration tree of all possible states, actions and their results.

Figure 5 illustrates state-transition planning system, where each node represents the state at some time $t$ and actions that are to be made to map the state at $t$ to the state at $t + 1$. State-transition function $\gamma$ gives the next state, or possible next states, after an action.
Let’s consider the following notations:

- $S_t$ – the state of enumeration scheme for a certain partial schedule in moment $t$;
- $A_s$ – the set of activities for the certain $s$-state;
- $\tilde{s}_{is}(\cdot), \tilde{c}_{is}(\cdot)$ – the start and finish interval times of activity $i$,

where $\tilde{c}_i(\cdot) = \tilde{s}_i(\cdot) + \tilde{d}_i(\cdot)$;
- $\tilde{d}_i(\cdot)$ – fuzzy interval duration of activity $i$;
- $Planneds$ – set of already scheduled activities for the $s$-state;
- $Readys$ – set of suitable (active) to scheduling activities for the $s$-state with respect to precedence relations;
- $e_i(\cdot)$ – earliest precedence-feasible starting time of activity $i$;
- $\text{Pred}(i)$ and $\text{Succ}(i)$ – the set of predecessor and successor activities for the $i$-activity

Having a discrete sequence of states, time constraints, sequence order and a number of planning activities we can translate our scheduling problem into the computational planning algorithm for the $s$-states as follows:

**Step 1: Initialization.**

Set $s_1 = 1$; Dummy activity $i_1 = 0$; $\bar{s} := 0$; $\bar{s}_i = 0$; $Planneds_1 = \emptyset$;

**Step 2: Updating sets of activities.**

Increase state number $s = s + 1$.

Update set of scheduled activities: $Planneds = Planneds_{s-1} \cup \{i_{s-1}\}$.

Compose the set of suitable activities:

$Readys = \{i \in A \setminus Planneds | \text{pred}(i) \subseteq Planneds\}$.

If the last dummy activity is active, then $n + 1 \in Readys$, then store the current solution and go to Step 5. Else go to step 3.

**Step 3: Selecting the next activity from the set Readys to be scheduled.**

If there is no untested activity left in the set Readys then go to step 5,

Else select the next activity $i_s \in Readys$.

**Step 4: Computing the earliest precedence feasible starting time.**

Compute the earliest precedence feasible starting time of the next activity:

$e_i(\cdot) = \max\{\tilde{c}_i(\cdot) | i \in \text{pred}(i_s)\} + 1$. Go to Step 2.

---

**Fig. 5.** The classical state-oriented dynamic system illustrated as an unfolded computational graph.
Step 5: Backtracking.

Decrease the precedence state by 1: $s = s - 1$. If the precedence state $s = 1$, then STOP.
Else go to Step 4.

For each state of enumeration scheme evaluates all possible partial schedules and store the current solution until all the activities are scheduled. A detailed description of multi-mode resource-constrained project scheduling problems with non-preemptive activity splitting and the correspondent algorithms were given in [18].

5 Conclusion

In this work a new descriptive and operational model for temporal reasoning and temporal knowledge representation of relations between activities in computational graph have been introduced. In many situations involving computations with uncertain fuzzy variables, fuzzy arithmetic and a number of computations should be made to obtain feasible/optimal solution. In this paper we treat fuzzy intervals as crisp intervals of gradual numbers. A state-transition model for a planning system was introduced and the idea to apply unfolded computational graph to a planning system was proposed. On a future research, this approach could be used in topology: the unfolding of the system represents in a single branching structure all its possible computations, for example, in fuzzy graph transformation systems. The topological distribution of a system thus can be represented by a graph structure and the temporal dynamics of the system.

Acknowledgments. The reported study was funded by the Russian Foundation for Basic Research according to the research project #20-01-00197.

References

1. Allen, J.: Towards a general theory of action and time. Artif. Intell. 23, 123–154 (1984). https://doi.org/10.1016/0004-3702(84)90008-0
2. Allen, J., Hayes, P.: A common-sense theory of time. In: Joshi, A. (ed.) Proceedings of the Ninth International Joint Conference on Artificial Intelligence (IJCAI 1985), San Francisco, pp. 528–531 (1985)
3. Dubois, D., Kerre, E., Mesiar, R., Prade, H.: Fuzzy interval analysis. In: Dubois, D., Prade, H. (eds.) Fundamentals of Fuzzy Sets: The Handbooks of Fuzzy Sets Series. FSHS, vol. 7, pp. 483–581. Springer, Boston (2000). https://doi.org/10.1007/978-1-4615-4429-6_11
4. Ghallab, M., Nau, D., Traverso, P.: Automated Planning and Acting. Cambridge University Press, Cambridge (2016)
5. Milea, V., Frasincar, F., Kaymak, U.: tOWL: a temporal web ontology language. IEEE Trans. Syst. Man Cybern. Part B (Cybern.) 42(1), 268–281 (2012). https://doi.org/10.1109/TSMCB.2011.2162582
6. Fortin, J., Dubois, D., Fargier, H.: Gradual numbers and their application to fuzzy interval analysis. IEEE Trans. Fuzzy Syst. 16(2), 388–402 (2008). https://doi.org/10.1109/TFUZZ.2006.890680
7. Dubois, D., Prade, H.: Operations on fuzzy numbers. Int. J. Syst. Sci. 9(6), 613–626 (1978). https://doi.org/10.1080/00207727808941724
8. Dubois, D., Prade, H.: Gradual numbers and fuzzy solutions to fuzzy optimization problems. In: Pelta, D.A., Cruz Corona, C. (eds.) Soft Computing Based Optimization and Decision Models. SFSC, vol. 360, pp. 223–229. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-64286-4_13
9. Verdegay, J.L.: A dual approach to solve the fuzzy linear programming problem. Fuzzy Sets Syst. 14(2), 131–141 (1984). https://doi.org/10.1016/0165-0114(84)90096-4
10. Dubois, D., Fargier, H., Fortemps, P.: Fuzzy scheduling: modeling flexible constraints vs: coping with incomplete knowledge. Eur. J. Oper. Res. 147, 231–252 (2003). https://doi.org/10.1016/S0377-2217(02)00558-1
11. Dubois, D., Fargier, H., Fortin, J.: A generalized vertex method for computing with fuzzy intervals. In: Proceedings of the IEEE International Conference on Fuzzy Systems FUZZ-IEEE, pp. 541–546 (2004). https://doi.org/10.1109/fuzzy.2004.1375793
12. Dubois, D., Fargier, H., Galvagnon, V.: On latest starting times and floats in activity networks with ill-known durations. Eur. J. Oper. Res. 147, 266–280 (2003)
13. Zielinski, P.: On computing the latest starting times and floats of activities in a network with imprecise durations. Fuzzy Sets Syst. 150, 53–76 (2005). https://doi.org/10.1016/j.fss.2004.08.007
14. Fortin, J., Dubois, D.: Solving fuzzy PERT using gradual real numbers. In: Proceedings of the 3rd European Starting AI Researcher Symposium (STAIRS 2006), pp. 196–207 (2006)
15. Dubois, D., Prade, H.: Additions of interactive fuzzy numbers. IEEE Trans. Autom. Control 26(4), 99–135 (1981)
16. Knyazeva, M., Bozhenyuk, A., Kacprzyk, J.: Modeling decisions for project scheduling optimization problem based on type-2 fuzzy numbers. In: Batyrshin, I., Martinez-Villaseñor, M.L., Ponce Espinosa, H.E. (eds.) MICAI 2018. LNCS (LNAI), vol. 11288, pp. 357–368. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-04491-6_27
17. Lodwick, W.A., Untiedt, E.A.: A comparison of interval analysis using constraint interval arithmetic and fuzzy interval analysis using gradual numbers. In: NAFIPS 2008: Annual Meeting of the North American Fuzzy Information Processing Society, New York City, NY, pp. 1–6 (2008)
18. Cheng, J., Fowler, J., Kempf, K., Mason, S.: Multi-mode resource-constrained project scheduling problems with non-preemptive activity splitting. Comput. Oper. Res. 53, 275–287 (2015). https://doi.org/10.1016/j.cor.2014.04.018