Latest starting times and floats of activities in networks with uncertain durations*

Paweł Zieliński
Institute of Mathematics, Technical University of Opole
Luboszycka 3, 45-036 Opole, Poland
pziel@po.opole.pl

Abstract

The computation of latest starting times and floats of activities in networks with uncertain durations, represented by means of interval and fuzzy numbers, is investigated.

Keywords: Scheduling, intervals, fuzzy intervals, critical path analysis.

1 Introduction

The critical path method (CPM) is one of the most frequently used tools in Operations Research. It is applied to the analysis of complex projects from the point of view of the planning and control of their realization in time. The essence of the CPM is the representation of the project by an activity network, where activities with given duration times are related to each other by means of precedence constraints. Determining, in such a network, many project characteristics, with the most important such as: earliest and latest starting times of activities, floats of activities and the minimum project duration, is an important task in practice. When the durations of activities are precisely known, these project characteristics are easy to compute in the network by means of the CPM. In case of ill-known activity duration times, the problem becomes more complicated even if their estimations are modeled by intervals. Namely, floats can no longer be recovered from the intervals containing earliest and latest starting times, and critical paths may no longer exist. Several works tried to cope with this problem (see [2] for a survey). The first attempt to obtain a correct solution has been made in [1]. There has been provided the *possibilistic* representation of the problem of determining fuzzy latest starting times of activities and their floats, its difficulty has been pointed out, but without proposing any solution methods.

In this paper we investigate the determination of latest starting times and floats of activities in networks with duration intervals. Then we extend the results to networks with fuzzy duration times. So far, these problems have been completely solved when networks are series parallel (see [8], [6]). Here, we propose new algorithms for latest starting times in general networks. We also present complexity results for floats (the computation of floats is probably intractable) and describe some polynomially solvable cases.

2 Latest starting times and floats of activities in a network with duration intervals

A network $G=<V,A>$ being a project activity-on-arc model, is given. V is the set of events, $|V|=n$, and A is the set of activities, $|A|=m$. The network G is a directed, connected and acyclic graph. Activity durations are to be chosen from intervals $T_{ij}=[\underline{t}_{ij},\overline{t}_{ij}]$, $(i,j)\in A$, two nodes are distinguished as the initial and final node, respectively. We introduce some additional notations. Let T denote a configuration of activity durations $t_{ij}\in T_{ij}$, $(i,j)\in A$, while $t_{ij}(T)$ denotes the duration of activity (i,j) in configuration T. We use $\mathfrak{T}$ to denote the set of possible configurations of the activity durations, i.e. $\mathfrak{T}$ is the Cartesian product of corresponding intervals T_{ij}, $(i,j)\in A$. $SUCC(i,j)$ (resp. $PREC(i,j)$) stands for the set of all arcs that come after (resp. before) $(i,j)\in A$. $\underline{\mathfrak{T}}^s(x)$, $x\geq 0$, is the Cartesian product of time intervals $\underline{T}^s_{ij}(x)$, $(i,j)\in A$,

* This work was supported by grant no. 7T11F02120 from the State Committee for Scientific Research (Komitet Badań Naukowych).

given as follows:

$$\underline{T}^s_{ij}(x) = \begin{cases} [\underline{t}_{ij}, \overline{t}_{ij}] & \text{for} \quad (i,j) \in SUCC(k,l), \\ [x,x] & \text{for} \quad (i,j) = (k,l), \\ [\underline{t}_{ij}, \underline{t}_{ij}] & \quad\;\; \text{otherwise}. \end{cases}$$

and $\overline{\mathfrak{T}}^s(x)$ is the Cartesian product of $\overline{T}^s_{ij}(x)$, $(i,j) \in A$, given as follows:

$$\overline{T}^s_{ij}(x) = \begin{cases} [\underline{t}_{ij}, \overline{t}_{ij}] & \text{for} \quad (i,j) \in SUCC(k,l), \\ [x,x] & \text{for} \quad (i,j) = (k,l), \\ [\overline{t}_{ij}, \overline{t}_{ij}] & \quad\;\; \text{otherwise}. \end{cases}$$

We study two problems which have been originally stated in [8], [6]. The first one is that of determining the interval T^l_{kl} (bounds) of possible values of latest starting times t^l_{kl} for a given activity $(k,l) \in A$, i.e. the interval $T^l_{kl} = [\underline{t}^l_{kl}, \overline{t}^l_{kl}]$ formed by the $\underline{t}^l_{kl} = \min t^l_{kl}(T)$ and $\overline{t}^l_{kl} = \max t^l_{kl}(T)$, where min and max are taken over the set of possible configurations $\mathfrak{T}$. $t^l_{kl}(T)$ is the latest starting time of activity (k,l) in configuration T. The latest time $t^l_{kl}(T)$ is computed by means of the formula $t^l_{kl}(T) = t^l_l(T) - t_{kl}(T)$, where $t^l_l(T)$ is the latest moment of occurrence of event l in configuration T. The second problem is that of determining the interval F_{kl} (bounds) of possible values of floats f_{kl} for a given activity $(k,l) \in A$, i.e. the interval $F_{kl} = [\underline{f}_{kl}, \overline{f}_{kl}]$ formed by the $\underline{f}_{kl} = \min f_{kl}(T)$ and $\overline{f}_{kl} = \max f_{kl}(T)$, where min and max are taken over all possible configurations of the activity durations $\mathfrak{T}$. $f_{kl}(T)$ is the float of activity (k,l) in configuration T. Float $f_{kl}(T)$ is determined by means of the formula $f_{kl}(T) = t^l_l(T) - t^e_k(T) - t_{kl}(T)$, where $t^e_k(T)$ and $t^l_l(T)$ are the earliest and the latest moments of occurrence of events $k \in V$ and $l \in V$, respectively, in configuration T.

2.1 Determination of bounds on latest starting times of an activity

We first describe a problem closely related to the one of computing a lower bound on latest starting times of an activity: that of evaluating *possible criticality* of an activity.

An activity (a path) is *possibly critical* in G if and only if there exists a configuration of times $T \in \mathfrak{T}$, such that the activity (the path) is critical in the usual sense in G. The possible criticality have been thoroughly investigated in [3], [4]. The problem of the possible criticality for a path is polynomially solvable. Unfortunately, the one for an activity turned out to be strongly $\mathcal{NP}$-complete for general networks and remains $\mathcal{NP}$-complete even when a network is restricted to be planar (see [5]). In [8] a polynomial algorithm has been provided only in case of series-parallel networks. This problem is polynomially solvable for general networks, under the assumption that activities $(i,j) \in PRECC(k,l)$ or $(i,j) \in SUCCC(k,l)$ have precise duration times, where $(k,l) \in A$ is an activity whose possible criticality is evaluated.

Consider the case $(i,j) \in PRECC(k,l)$ have precise duration times. The logic of the algorithm (Algorithm 1), which can distinguish whether an activity (k,l) is possibly critical, is to construct a configuration $T \in \underline{\mathfrak{T}}^s(\overline{t}_{kl})$ in which (k,l) is critical in the usual sense. To find T, node labeling is performed with conveniently setting activity durations. If such a configuration is successfully determined then (k,l) is possibly critical, otherwise not. The running time of the algorithm is $O(m)$.

An algorithm, which can distinguish whether an activity (k,l) is possibly critical in G with activities $(i,j) \in SUCC(k,l)$ having precise duration times, is identical to Algorithm 1. It is enough to reverse arcs in network G and carry out the computations from node n down to 1.

We now present an algorithm for determining $\underline{t}^l_{kl}$ of a given activity $(k,l) \in A$. Let us recall an important result, given by Dubois et al. [6], that allows to reduce the set of configurations $\mathfrak{T}$.

Proposition 1. $\underline{t}^l_{kl} = \min_{T \in \underline{\mathfrak{T}}^s(\overline{t}_{kl})} t^l_{kl}(T)$. *Moreover, the minimum $t^l_{kl}(T)$ is attained on the vertices of the hyper-rectangle $\underline{\mathfrak{T}}^s(\overline{t}_{kl})$.*

The key lemma for constructing the algorithm for computing $\underline{t}^l_{kl}$ (Algorithm 2) is the following one.

Lemma 1. *Let f^*_{kl} be the minimal nonnegative real number such that (k,l) with a duration time $\overline{t}_{kl} + f^*_{kl}$ becomes possibly critical. Then $\underline{t}^e_k + f^*_{kl} = \min_{T \in \underline{\mathfrak{T}}^s(\overline{t}_{kl})} t^l_{kl}(T)$, where $\underline{t}^e_k$ is the earliest moment when event k occurs.*

Proof. Let us observe that $t^l_{kl}(T) = \underline{t}^e_k + f_{kl}(T)$ for all $T \in \underline{\mathfrak{T}}^s(\overline{t}_{kl})$. This follows from the fact that $t^e_k(T)$ is equal to $\underline{t}^e_k$ for all $T \in \underline{\mathfrak{T}}^s(\overline{t}_{kl})$. Thus to prove $\underline{t}^e_k + f^*_{kl} = \min_{T \in \underline{\mathfrak{T}}^s(\overline{t}_{kl})} t^l_{kl}(T)$, we only need to show that $f^*_{kl} = \min_{T \in \underline{\mathfrak{T}}^s(\overline{t}_{kl})} f_{kl}(T)$. Assume to the contrary that $f^*_{kl} \neq \min_{T\in\underline{\mathfrak{T}}^s(\bar{t}_{kl})} f_{kl}(T)$. Let us consider two cases. Case 1: $f^*_{kl} < \min_{T\in\underline{\mathfrak{T}}^s(\bar{t}_{kl})} f_{kl}(T)$. The result is $\min_{T\in\underline{\mathfrak{T}}^s(\bar{t}_{kl})} f_{kl}(T) > 0$ and consequently (k,l) with duration time $\bar{t}_{kl} + f^*_{kl}$ is not critical for all $T \in \underline{\mathfrak{T}}^s(\bar{t}_{kl} + f^*_{kl})$. This contradicts possible criticality of (k,l). Case 2: $f^*_{kl} > \min_{T\in\underline{\mathfrak{T}}^s(\bar{t}_{kl})} f_{kl}(T)$. Hence there exists a configuration $T^{'} \in \underline{\mathfrak{T}}^s(\bar{t}_{kl})$ such that $f^*_{kl} > f_{kl}(T^{'})$. Let us increase duration time of (k,l) from $\bar{t}_{kl}$ to $\bar{t}_{kl} + f_{kl}(T^{'})$ in $T^{'}$. For this new configuration, say $T^{''}$, $T^{''} \in \underline{\mathfrak{T}}^s(\bar{t}_{kl} + f_{kl}(T^{'}))$, (k,l) is critical and therefore it is possibly critical. This contradicts the assumption that f^*_{kl} is the minimal number such that (k,l) with duration time $\bar{t}_{kl} + f^*_{kl}$ becomes possibly critical. □

The idea of Algorithm 2 is based on Lemma 1. It consists in finding the minimal nonnegative real number that added to the upper bound of duration interval of a specified (k,l) makes it possibly critical. In each iteration of the algorithm the duration time of (k,l) is conveniently increased (row 9) and possible criticality of (k,l) for such an increased duration is tested. The testing is reduced to applying the algorithm for asserting the possible criticality of (k,l) (Algorithm 1). This process is repeated until the activity becomes possibly critical. Then from Lemma 1, we immediately obtain the minimal latest starting time of (k,l). It is worth noticing that in row 11 only PHASE 2 of Algorithm 1 is called, because the earliest moments of occurrence of events t^e_i, node labels $label(i)$, and duration times t_{ij}, for $i, j \leq l$, computed in the first call (row 2) remain unchanged. The entire complexity of the algorithm is $O(mn)$.

We now pass on to the problem of computing an upper bound on latest starting times of an activity. We first examine a problem closely related to it: that of evaluating *necessary criticality* of an activity.

An activity (a path) is *necessarily critical* in G if and only if for every configuration of times $T \in \mathfrak{T}$, the activity (the path) is critical in G in the usual sense. See [2] for a detailed study of the necessary criticality. The problem of the necessary criticality for a path can be solved in polynomial time. Unfortunately, the one for an activity does not seem to be such. The question of proving this fact is still open. In [8] a polynomial algorithm has been provided only for series-parallel networks. However, this problem is polynomially solvable for general networks, under the assumption that activities $(i,j) \in PRECC(k,l)$ or $(i,j) \in SUCCC(k,l)$ have precise duration times, where $(k,l) \in A$ is an activity whose necessary criticality is evaluated.

Consider the case $(i,j) \in PRECC(k,l)$ have precise duration times. The main idea of an algorithm (Algorithm 3) which can evaluate the necessary criticality of activity (k,l) is to find a configuration $T \in \overline{\mathfrak{T}}^s(\underline{t}_{kl})$ in which (k,l) is not critical in the usual sense. If such a configuration T is successfully determined then (k,l) is not necessarily critical, otherwise it is. The algorithm is similar in spirit to the one for the possible criticality of an activity. The running time of the algorithm is $O(m)$.

An algorithm, which can assert whether an activity (k,l) is necessarily critical in G with activities $(i,j) \in SUCC(k,l)$ having precise duration times is identical to Algorithm 3. It is sufficient to reverse arcs in network G and carry out the computations from node n down to 1.

Here, we describe an algorithm for computing the maximal latest starting time of an activity. Let us recall an important result, given by Dubois et al. [6], that allows to reduce the set of configurations $\mathfrak{T}$.

Proposition 2. $\bar{t}^l_{kl} = \max_{T\in\overline{\mathfrak{T}}^s(\underline{t}_{kl})} t^l_{kl}(T)$. *Moreover, the maximum $t^l_{kl}(T)$ is attained on the vertices of the hyper-rectangle $\overline{\mathfrak{T}}^s(\underline{t}_{kl})$.*

The construction of the algorithm for determining $\bar{t}^l_{kl}$ of a given activity $(k,l) \in A$ (Algorithm 4) is based on the following lemma.

Lemma 2. *Let f^*_{kl} be the minimal nonnegative real number such that (k,l) with a duration time $\underline{t}_{kl} + f^*_{kl}$ becomes necessarily critical. Then $\bar{t}^e_k + f^*_{kl} = \max_{T\in\overline{\mathfrak{T}}^s(\underline{t}_{kl})} t^l_{kl}(T)$, where $\bar{t}^e_k$ is the earliest moment when event k occurs.*

Proof. Our proof starts with observation that $t^l_{kl}(T) = \bar{t}^e_k + f_{kl}(T)$, for all $T \in \overline{\mathfrak{T}}^s(\underline{t}_{kl})$. The observation follows from the fact that $t^e_k(T) = \bar{t}^e_k$, for all $T \in \overline{\mathfrak{T}}^s(\underline{t}_{kl})$. Hence in order to prove $\bar{t}^e_k + f^*_{kl} = \max_{T\in\overline{\mathfrak{T}}^s(\underline{t}_{kl})} t^l_{kl}(T)$, it suffices to show that $f^*_{kl} = \max_{T\in\overline{\mathfrak{T}}^s(\underline{t}_{kl})} f_{kl}(T)$. Suppose on the contrary that $f^*_{kl} \neq \max_{T\in\overline{\mathfrak{T}}^s(\underline{t}_{kl})} f_{kl}(T)$. Then we should consider the following two case. Case 1: $f^*_{kl} < \max_{T\in\overline{\mathfrak{T}}^s(\underline{t}_{kl})} f_{kl}(T)$. This implies that there exists a configuration $T^{'} \in \overline{\mathfrak{T}}^s(\underline{t}_{kl})$ such that $f^*_{kl} < f_{kl}(T^{'})$, which gives $f_{kl}(T^{'}) > 0$. Consequently

(k,l) is not critical in T'. Let us increase the duration time of (k,l) from $\underline{t}_{kl}$ to $\underline{t}_{kl} + f^*_{kl}$ in T'. For this new configuration, say T'', $T'' \in \overline{\mathfrak{T}}^s(\underline{t}_{kl} + f^*_{kl})$, (k,l) is still not critical, which contradicts the assumption that (k,l) is critical for all $T \in \overline{\mathfrak{T}}^s(\underline{t}_{kl} + f^*_{kl})$. Case 2: $f^*_{kl} > \max_{T \in \overline{\mathfrak{T}}^s(\underline{t}_{kl})} f_{kl}(T)$. Thus (k,l) is critical with duration time $\underline{t}_{kl} + \max_{T \in \overline{\mathfrak{T}}^s(\underline{t}_{kl})} f_{kl}(T)$, for all $T \in \overline{\mathfrak{T}}^s(\underline{t}_{kl} + \max_{T \in \overline{\mathfrak{T}}^s(\underline{t}_{kl})} f_{kl}(T))$, and therefore it is necessarily critical. This contradicts the assumption that f^*_{kl} is the minimal number such that (k,l) becomes necessarily critical. □

The main idea of Algorithm 4 is based on Lemma 2. It consists in determining the minimal nonnegative real number that added to the lower bound of duration interval of a specified activity (k,l) makes it necessarily critical. Namely, in each iteration the duration time of (k,l) is suitably increased (row 9) and necessary criticality of (k,l) for such an increased duration is evaluated. The evaluation boils down to applying Algorithm 3. This process is repeated until the activity becomes necessarily critical. Then Lemma 2 gives the maximal latest starting time of (k,l). It is worth noticing that in row 11 only PHASE 2 of Algorithm 3 is called, because the earliest moments of occurrence of events t^e_i, node labels $label(i)$, and duration times t_{ij}, for $i,j \le l$, computed in the first call (row 2) remain unchanged. The complexity of the algorithm is $O(mn)$.

2.2 Determination of bounds on floats of an activity

Here, we consider the problem of determining bounds on floats, $\underline{f}_{kl}$ and $\overline{f}_{kl}$, of a given activity $(k,l) \in A$. There are obvious connections between the notions of criticality and bounds on floats. An activity $(k,l) \in A$ is possibly (resp. necessarily) critical in G if and only if $\underline{f}_{kl} = 0$ (resp. $\overline{f}_{kl} = 0$). Accordingly, the problem of computing $\underline{f}_{kl}$ is strongly $\mathcal{NP}$-hard for general networks and remains $\mathcal{NP}$-hard even for planar networks (see [5]). This negative result encourages us to look for approximation algorithms. It has turned out that if $\mathcal{P} \neq \mathcal{NP}$ then it is not possible to approximate $\underline{f}_{kl}$ within a factor smaller than 1, even when G is restricted to a planar network. Unfortunately, the question, still unanswered, is whether the problem of computing $\overline{f}_{kl}$ is polynomially solvable. In [6] some heuristic methods have been proposed. At present, these problems are effectively solvable in series-parallel networks (see [8]). However, under the assumption that activities $(i,j) \in PRECC(k,l)$ or $(i,j) \in SUCCC(k,l)$ have precise duration times, the problems can be exactly solved in a polynomial time for general networks.

Lemma 3. *Assume that $(i,j) \in PREC(k,l)$ have precise duration times and f^*_{kl} is the minimal nonnegative real number such that (k,l) with a duration time $\overline{t}_{kl} + f^*_{kl}$ (resp. $\underline{t}_{kl} + f^*_{kl}$) becomes possibly (resp. necessary) critical. Then $\underline{f}_{kl} = f^*_{kl}$ (resp. $\overline{f}_{kl} = f^*_{kl}$).*

Proof. It suffices to show that $f^*_{kl} = \min_{T \in \underline{\mathfrak{T}}^s(\overline{t}_{kl})} f_{kl}(T)$ (resp. $f^*_{kl} = \max_{T \in \overline{\mathfrak{T}}^s(\underline{t}_{kl})} f_{kl}(T)$). The proof of this equality runs in the same manner as the one of Lemma 1 (resp. Lemma 2). □

To find the number f^*_{kl} in Lemma 3, it is sufficient to apply Algorithm 2 and Algorithm 4 for computing bounds on latest starting times of (k,l). From Lemmas 1 and 2 it follows that they implicitly compute f^*_{kl} to determine bounds on latest starting times. Thus, we obtain $O(mn)$ algorithms for determining bounds on floats.

If activities $(i,j) \in SUCC(k,l)$ have precise duration times, algorithms for bounds on floats are similar. It is sufficient to reverse arcs in network G and apply Algorithm 2 and Algorithm 4.

3 Latest starting times and floats of activities in a network with fuzzy durations

Now we focus on the fuzzy case. All the elements of the network G are the same as in the interval case except for activity duration times, which are determined by means of fuzzy numbers $\tilde{T}_{ij}$, $(i,j) \in A$, which imprecisely determine duration times of activities $(i,j) \in A$. Fuzzy number $\tilde{T}_{ij}$ expresses uncertainty connected with the ill-known activity duration time modeled by this number. It generates possibility distribution for sets of values containing the unknown activity duration. More formally, we say that the assertion of the form "t_{ij} is $\tilde{T}_{ij}$", where t_{ij} is a variable and $\tilde{T}_{ij}$ is a fuzzy number, generates the possibility distribution of t_{ij} with respect to the following formula (see [6],[7]): $\Pi(\mathrm{t}_{ij} = x) = \mu_{\tilde{T}_{ij}}(x)$, $x \in \mathbb{R}_+$.

Let T be a configuration of activity duration times in the network with activity times $t_{ij} \in \mathbb{R}_+$, $(i,j) \in A$. Thus, the (joint) possibility distribution over configurations, induced by the $\tilde{T}_{ij}$'s is $\pi(T) = \min_{(i,j)\in A} \mu_{\tilde{T}_{ij}}(t_{ij})$, $T \in \mathbb{R}^m_+$. Hence, the possibility distribution describing possible values for latest starting times $\mathfrak{t}^l_{kl}$ (resp. float $\mathfrak{f}_{kl}$) of an activity (k,l) is defined in following way (see [6]):

$$\mu_{\tilde{T}^l_{kl}}(x) = \Pi(\mathfrak{t}^l_{kl} = x) = \sup_{T: x = t^l_{kl}(T)} \pi(T), \ \ x \in \mathbb{R}_+,$$
$$\mu_{\tilde{F}_{kl}}(x) = \Pi(\mathfrak{f}_{kl} = x) = \sup_{T: x = f_{kl}(T)} \pi(T), \ \ x \in \mathbb{R}_+,$$

where $t^l_{kl}(T)$ (resp. $f_{kl}(T)$) is the latest starting time (resp. the float) of (k,l) in configuration T.

The above fuzzy quantities (possibility distributions) can be determined via the use of α-cuts. That is, a method (in the interval case) computes α-cuts, $\tilde{T}^l_{kl}(\alpha)$ and $\tilde{F}_{kl}(\alpha)$, of each fuzzy latest starting time $\tilde{T}^l_{kl}$ and float $\tilde{F}_{kl}$ in a network with duration intervals $\tilde{T}_{ij}(\alpha) = [\underline{t}_{ij}(\alpha), \overline{t}_{ij}(\alpha)]$, $(i,j) \in A$. Then the fuzzy quantities, $\tilde{T}^l_{kl}$ and $\tilde{F}_{kl}$, are reconstructed from their α-cuts. This approach makes sense since intervals $\tilde{T}^l_{kl}(\alpha) = [\underline{t}^l_{kl}(\alpha), \overline{t}^l_{kl}(\alpha)]$ and $\tilde{F}_{kl}(\alpha) = [\underline{f}_{kl}(\alpha), \overline{f}_{kl}(\alpha)]$ are nested. Such an approach has been proposed in [8], [6] and corresponding polynomial algorithms have been provided for networks having a special topology, namely series-parallel ones. It is worth pointing out that the main difficulty of determining fuzzy project characteristics, when fuzzy numbers represent ill-known processing times, does not lie in the introduction of fuzzy sets. It is already present when only usual intervals are involved. Solving the interval valued case is the main difficulty. Thus, fuzzy latest starting time $\tilde{T}^l_{kl}$, i.e. its α-cuts, in general networks can be determined by means of the algorithm for computing bounds on latest starting times (Algorithm 2 and Algorithm 4), described in Section 2.1.

As far as the determination of fuzzy float $\tilde{F}_{kl}$ is concerned, the problem does not seem to be easy, because it is more general than the one, in the interval case, of computing bounds on floats. Hence, the determination of $\tilde{F}_{kl}$ is $\mathcal{NP}$-hard. However, in some cases one can determine $\tilde{F}_{kl}$, that is in cases in which computing bounds on floats is polynomially solvable (see Section 2.2).

References

[1] J.J. Buckley, Fuzzy PERT, in: *Applications of Fuzzy Set Methodologies in Industrial Engineering*, ed. by G.W. Evans, W. Karwowski and M.R. Wilhelm, Elsevier, Amsterdam-Oxford-New York-Tokyo 1989, 103–114.

[2] S. Chanas, D. Dubois, P. Zieliński, On the sure criticality of tasks in activity networks with imprecise durations, *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics* 32 (2002) 393-407.

[3] S. Chanas, P. Zieliński, Critical path analysis in the network with fuzzy activity times, *Fuzzy Sets and Systems* 122 (2001) 195–204.

[4] S. Chanas, P. Zieliński, The computational complexity of the criticality problems in a network with interval activity times, *European Journal of Operational Research* 136 (3) (2002) 541–550.

[5] S. Chanas, P. Zieliński, On the hardness of evaluating criticality of activities in a planar network with duration intervals, *Operations Research Letters* 31 (2003) 53-59.

[6] D. Dubois, H. Fargier, V. Galvagnon, On latest starting times and floats in activity networks with ill-known durations, *European Journal of Operational Research* 147 (2003) 266–280.

[7] D. Dubois, H. Fargier, P. Fortemps, Fuzzy scheduling: Modelling flexible constraints vs. coping with incomplete knowledge *European Journal of Operational Research* 147 (2003) 231-252.

[8] H. Fargier, V. Galvagnon, D. Dubois, Fuzzy PERT in series-parallel graphs, *9-th IEEE Int. Conf. on Fuzzy Systems*, San Antonio, TX, 2000 717–722.

Algorithm 1 Asserting whether an activity is possibly critical

Require: A network $G =< V,A >$, a specified activity $(k,l) \in A$, time intervals $T_{ij} = [\underline{t}_{ij}, \overline{t}_{ij}]$, $(i,j) \in A$.
Ensure: A configuration $T \in \underline{\mathfrak{T}}^s(\overline{t}_{kl})$, $PossCritical = true$ if (k,l) is possibly critical, $false$ otherwise.

$\triangleright$ PHASE 1:
$t_1^e \leftarrow 0$; $label(1) \leftarrow false$;
for $j \leftarrow 2$ to $l-1$ **do**
  for all $i \in Prec(j)$ **do**
    $t_{ij} \leftarrow \underline{t}_{ij}$
  end for
  $t_j^e \leftarrow \max\{t_i^e + t_{ij} \,|\, i \in Prec(j)\}$; $label(j) \leftarrow false$
end for
for all $i \in Prec(l)$ **do**
  if $i \neq k$ **then**
    $t_{il} \leftarrow \underline{t}_{il}$
  end if
end for
$t_{kl} \leftarrow \overline{t}_{kl}$;
$t_l^e \leftarrow \max\{t_i^e + t_{il} \,|\, i \in Prec(l)\}$; $label(l) \leftarrow false$;
if $t_l^e \neq t_k^e + t_{kl}$ **then**
  $PossCritical \leftarrow false$; **exit**
end if
$\triangleright$ PHASE 2:
$label(l) \leftarrow true$;
for $j \leftarrow l+1$ to n **do**
  for all $i \in Prec(j)$ **do**
    if $label(i) = true$ **then**
      $t_{ij} \leftarrow \overline{t}_{ij}$
    else
      $t_{ij} \leftarrow \underline{t}_{ij}$
    end if
  end for
  $t_j^e \leftarrow \max\{t_i^e + t_{ij} \,|\, i \in Prec(j)\}$;
  if $\{i \,|\, i \in Prec(j), label(i) = true, t_j^e = t_i^e + t_{ij}\} \neq \emptyset$ **then**
    $label(j) \leftarrow true$
  else
    $label(j) \leftarrow false$
  end if
end for
$PossCritical \leftarrow label(n)$;

Algorithm 2 Computing the minimal latest starting time of an activity

Require: A network $G =< V,A >$, a specified activity $(k,l) \in A$, time intervals $T_{ij} = [\underline{t}_{ij}, \overline{t}_{ij}]$, $(i,j) \in A$
Ensure: The minimal latest starting time of (k,l), $\underline{t}_{kl}^l$.

$f_{kl} \leftarrow 0$;
$\triangleright$ *Check possible criticality of* (k,l)..
call Algorithm 1;
while not *PossCritical* **do**
  if $label(l)$ **then**
    $\Delta \leftarrow \min\{t_j^e - t_i^e - t_{ij} \,|\, (i,j) \in A, label(i) = true, label(j) = false\}$
  else
    $\Delta \leftarrow t_l^e - t_k^e - t_{kl}$
  end if
  $t_l^e \leftarrow t_l^e + \Delta$;
  $f_{kl} \leftarrow f_{kl} + \Delta$;
  $\triangleright$ *Check possible criticality of* (k,l) *with implicitly*
  $\triangleright$ *increased duration* $\overline{t}_{kl} + f_{kl}$.
  call only PHASE 2 of Algorithm 1
end while
$\underline{t}_{kl}^l \leftarrow t_k^e + f_{kl}$; $\triangleright$ f_{kl} *equals* f_{kl}^*

Algorithm 3 Asserting whether an activity is necessarily critical

Require: A network $G =< V,A >$, a specified activity $(k,l) \in A$, time intervals $T_{ij} = [\underline{t}_{ij}, \overline{t}_{ij}]$, $(i,j) \in A$.

Ensure: A configuration $T \in \overline{\mathfrak{T}}^{s}(\underline{t}_{kl})$, $NecCritical = true$ if (k,l) is necessarily critical $false$ otherwise.

$\triangleright$ PHASE 1:

$t_1^e \leftarrow 0$; $label(1) \leftarrow true$;
for $j \leftarrow 2$ to $l-1$ **do**
    for all $i \in Prec(j)$ **do**
        $t_{ij} \leftarrow \overline{t}_{ij}$
    end for
    $t_j^e \leftarrow \max\{t_i^e + t_{ij} \,|\, i \in Prec(j)\}$; $label(j) \leftarrow true$
end for
for all $i \in Prec(l)$ **do**
    if $i \neq k$ **then**
        $t_{il} \leftarrow \overline{t}_{il}$
    end if
end for
$t_{kl} \leftarrow \underline{t}_{kl}$;
$t_l^e \leftarrow \max\{t_i^e + t_{il} \,|\, i \in Prec(l)\}$; $label(l) \leftarrow true$;
if $t_l^e \neq t_k^e + t_{kl}$ **then**
    $NecCritical \leftarrow false$; **exit**
end if

$\triangleright$ PHASE 2:

$label(l) \leftarrow false$;
for $j \leftarrow l+1$ to n **do**
    for all $i \in Prec(j)$ **do**
        if $label(i) = true$ **then**
            $t_{ij} \leftarrow \overline{t}_{ij}$
        else
            $t_{ij} \leftarrow \underline{t}_{ij}$
        end if
    end for
    $t_j^e \leftarrow \max\{t_i^e + t_{ij} \,|\, i \in Prec(j)\}$;
    if $\{i \,|\, i \in Prec(j), label(i) = false, t_j^e = t_i^e + t_{ij}\} \neq \emptyset$ **then**
        $label(j) \leftarrow false$
    else
        $label(j) \leftarrow true$
    end if
end for
$NecCritical \leftarrow$ **not** $label(n)$;

Algorithm 4 Computing the maximal latest starting time of an activity

Require: A network $G =< V,A >$, a specified activity $(k,l) \in A$, time intervals $T_{ij} = [\underline{t}_{ij}, \overline{t}_{ij}]$, $(i,j) \in A$

Ensure: The maximal latest starting time of (k,l), $\overline{t}_{kl}^{l}$.

$f_{kl} \leftarrow 0$;
$\triangleright$ *Check necessary criticality of* (k,l).
call Algorithm 3;
while not $NecCritical$ **do**
    if $label(l)$ **then**
        $\Delta \leftarrow t_l^e - t_k^e - t_{kl}$
    else
        $\Delta \leftarrow \min\{t_j^e - t_i^e - t_{ij} | (i,j) \in A, label(i) = false, label(j) = true\}$
    end if
    $t_l^e \leftarrow t_l^e + \Delta$;
    $f_{kl} \leftarrow f_{kl} + \Delta$;
    $\triangleright$ *Check necessary criticality of* (k,l) *with implicitly*
    $\triangleright$ *increased duration* $\underline{t}_{kl} + f_{kl}$.
    call only PHASE 2 of Algorithm 3
end while
$\overline{t}_{kl}^{l} \leftarrow t_k^e + f_{kl}$; $\triangleright$ f_{kl} *equals* f_{kl}^{*}