



ELSEVIER

Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

Fuzzy Sets and Systems 150 (2005) 53–76

FUZZY
sets and systems

www.elsevier.com/locate/fss

On computing the latest starting times and floats of activities in a network with imprecise durations

Paweł Zieliński

Institute of Mathematics, Wrocław University of Technology, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland

Received 19 December 2003; received in revised form 8 June 2004; accepted 23 August 2004

Available online 15 September 2004

Abstract

The paper deals with the problems of computing the intervals of possible values of the latest starting times and floats of activities in networks with imprecise durations, represented by means of interval or fuzzy numbers. So far, these problems have been completely solved when the networks are series parallel. We propose new polynomial algorithms for determining the intervals of the latest starting times in general networks. We also present some complexity results for floats (the computation of floats is probably intractable) and describe some polynomially solvable cases. Then we extend the results to networks with fuzzy durations.

© 2004 Elsevier B.V. All rights reserved.

Keywords: Project management and scheduling; Data intervals; Critical path analysis

1. Introduction

In this paper we wish to investigate the problems of computing the intervals (bounds) of possible values of the latest starting times and floats of activities in networks with uncertain durations modeled by fuzzy or interval numbers. These problems have attracted a considerable attention since the late 1970s, particularly because of their importance in project scheduling. So far, attempts to solve the problems have been mainly based on the critical path method (CPM) [16] with formulas for the forward and the backward recursions, where the crisp arithmetic is replaced with the interval (fuzzy) arithmetic [4,10,13,17,18,20]. For such straightforward extension of CPM method, it turns out that the forward recursion correctly computes the sets of possible values of the earliest starting times of activities but the backward recursion,

E-mail address: pziel@im.pwr.wroc.pl (P. Zieliński).

0165-0114/\$ - see front matter © 2004 Elsevier B.V. All rights reserved.

doi:10.1016/j.fss.2004.08.007

with interval minimum and subtraction, fails to compute the sets of possible values of the latest starting times of activities. So, the intervals of floats can no longer be recovered from the intervals containing the earliest and latest starting times. The first attempt to obtain a correct solution has been made in [2]. There has been provided a *possibilistic* representation of the problem of determining the fuzzy latest starting times of activities and their floats, a difficulty connected to it has been pointed out, but without proposing any solution methods. At present these problems have been completely solved when the networks are series parallel (see [11,9]). There have been provided $\mathcal{O}(n)$ algorithms for both problems. In [9] some heuristic methods for computing the intervals of possible values of the latest starting times and floats of activities in general networks have been proposed. Here, we give $\mathcal{O}(mn)$ algorithms for computing the intervals of possible values of the latest starting times of activities in general networks. We also present complexity results for floats (the computation of floats is probably intractable) and describe some polynomially solvable cases.

The paper is organized as follows. Section 2 formally defines the problems that we consider. Section 2.1 discusses some difficulties in the interval (fuzzy) CPM. Section 2.2 gives an $\mathcal{O}(mn)$ time algorithm for computing the lower bound on the latest starting times of a single activity in a general network. Section 2.3 gives an $\mathcal{O}(mn)$ time algorithm for computing the upper bound on the latest starting times of a single activity in a general network. Section 2.4 shows that the problem of determining the bounds on the float of an activity cannot be approximated within a factor smaller than 1 and describes some polynomially solvable cases. Section 3 extends the results to the networks with fuzzy durations. Section 4 concludes the paper.

2. The latest starting times and floats of activities in a network with duration intervals

A network $G = \langle V, A \rangle$, being a project activity-on-arc model, is given. V is the set of nodes (events), $|V| = n$, and A is the set of arcs (activities), $|A| = m$. The network G is a directed, connected and acyclic graph. The set $V = \{1, 2, \dots, n\}$ is labeled in such a way that $i < j$ for each activity $(i, j) \in A$. Weights of the arcs (activity durations) $(i, j) \in A$ are to be chosen from intervals $D_{ij} = [\underline{d}_{ij}, \bar{d}_{ij}]$, $\underline{d}_{ij} \geq 0$, two nodes 1 and n are distinguished as the initial and final node, respectively.

We introduce some additional notations.

- Ω denotes a configuration of activity durations $d_{ij} \in D_{ij}$, $(i, j) \in A$, while $d_{ij}(\Omega)$ denotes the duration of activity (i, j) in configuration Ω .
- \mathfrak{D} is the set of possible configurations of the activity durations, i.e. \mathfrak{D} is the Cartesian product of corresponding intervals D_{ij} , $(i, j) \in A$.
- P is the set of all paths in G from node 1 to node n .
- $P(k, l)$ is the set of all paths in G from node 1 to node n , which use activity $(k, l) \in A$.
- $l_p(\Omega)$ denotes the length of a path $p \in P$ in configuration Ω , $l_p(\Omega) = \sum_{(i,j) \in p} d_{ij}(\Omega)$.
- $Succ(i)$ (resp. $Prec(i)$) refers to the set of nodes that immediately follow (resp. precede) node $i \in V$.
- $SUCC(i, j)$ (resp. $PREC(i, j)$) stands for the set of all arcs that come after (resp. before) $(i, j) \in A$.

We study two problems which have been originally stated in [11,9].

The first one is that of determining the interval T_{kl}^l (bounds) of possible values of the latest starting times t_{kl}^l for a given activity $(k, l) \in A$, i.e. the interval $T_{kl}^l = [t_{kl}^l, \bar{t}_{kl}^l]$ formed by the $t_{kl}^l = \min t_{kl}^l(\Omega)$ and $\bar{t}_{kl}^l = \max t_{kl}^l(\Omega)$, where min and max are taken over the set of possible configurations \mathfrak{D} . $t_{kl}^l(\Omega)$ is

the latest starting time of activity (k, l) in configuration Ω . The latest time $t_{kl}^l(\Omega)$ is computed by means of the formula $t_{kl}^l(\Omega) = t_l^l(\Omega) - d_{kl}(\Omega)$, where $t_l^l(\Omega)$ is the latest moment of occurrence of event l in configuration Ω .

The second problem is that of determining the interval F_{kl} (bounds) of possible values of floats (total 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 $\underline{f}_{kl} = \min_{\Omega \in \mathfrak{D}} f_{kl}(\Omega)$ and $\overline{f}_{kl} = \max_{\Omega \in \mathfrak{D}} f_{kl}(\Omega)$, where min and max are taken over all possible configurations of the activity durations \mathfrak{D} . $f_{kl}(\Omega)$ is the float of activity (k, l) in configuration Ω . Float $f_{kl}(\Omega)$ is determined by means of the formula $f_{kl}(\Omega) = t_l^l(\Omega) - t_k^e(\Omega) - d_{kl}(\Omega)$, where $t_k^e(\Omega)$ and $t_l^l(\Omega)$ are the earliest and the latest moments of occurrence of events $k \in V$ and $l \in V$, respectively, in configuration Ω .

In view of above formulations, we may define four subproblems.

PLBLST: Compute the lower bound on the latest starting time \underline{t}_{kl}^l of a given activity $(k, l) \in A$, $\underline{t}_{kl}^l = \min_{\Omega \in \mathfrak{D}} t_{kl}^l(\Omega)$.

PUBLST: Compute the upper bound on the latest starting time \overline{t}_{kl}^l of a given activity $(k, l) \in A$, $\overline{t}_{kl}^l = \max_{\Omega \in \mathfrak{D}} t_{kl}^l(\Omega)$.

PLBF: Compute the lower bound on float \underline{f}_{kl} of a given activity $(k, l) \in A$, $\underline{f}_{kl} = \min_{\Omega \in \mathfrak{D}} f_{kl}(\Omega)$.

PUBF: Compute the upper bound on float \overline{f}_{kl} of a given activity $(k, l) \in A$, $\overline{f}_{kl} = \max_{\Omega \in \mathfrak{D}} f_{kl}(\Omega)$.

When we say an activity $(k, l) \in A$ is *critical in the usual sense* (or shortly *critical*) for configuration $\Omega \in \mathfrak{D}$, we mean that it has float $f_{kl}(\Omega) = 0$ in G after replacing the time intervals D_{ij} by exact values $d_{ij}(\Omega)$, $(i, j) \in A$. Similarly, a path $p \in P$ is *critical in the usual sense* for configuration $\Omega \in \mathfrak{D}$ if and only if all activities belonging to p are critical in the usual sense for configuration Ω (or equivalently p is the longest path in G for configuration Ω).

2.1. Difficulties in the interval CPM

In this section we indicate some difficulties in the interval CPM. However, we emphasize that all the considerations are also valid for the fuzzy CPM. When we say the interval (fuzzy) CPM, we mean the classical CPM method with formulas for the forward and the backward recursions, where the crisp arithmetic is replaced with the interval (fuzzy) arithmetic. For such extension of CPM method, it turns out that the forward recursion correctly computes the sets of possible values of the earliest moments when the events may occur [4,10,13] but the backward recursion fails to compute the sets of possible values of the latest moments when the events may occur and latest starting times of activities [17,18,20]. These times turn more and more imprecise while getting closer to the end of the calculation. Sometimes these times may be interval numbers with negative elements. Thus, the sets of possible values of floats cannot be obtained from the earliest and latest moments when the events may occur and the interval durations. This anomaly is caused by an interactivity of interval variables.

Let us consider an one activity network with interval duration $D_{12} = [1, 2]$. We obtain several dependencies among the interval variables, which follow from the forward recursion

$$\begin{aligned} T_1^e &= [0, 0], \\ T_2^e &= T_1^e \oplus D_{12} = [1, 2], \end{aligned} \tag{1}$$

where T_1^e and T_2^e are the intervals of possible values of the earliest moments when events 1 and 2 may occur, respectively, and from the backward recursion

$$T_2^l = T_2^e = [1, 2], \quad (2)$$

$$T_1^l = T_{12}^l = T_2^l \ominus D_{12} = [-1, 1], \quad (3)$$

$$F_{12} = T_2^l \ominus D_{12} \ominus T_1^e = [-1, 1], \quad (4)$$

where T_1^l and T_2^l are the intervals of possible values of the latest moments when events 1 and 2 may occur, respectively. Because of assignment (2) variable T_2^l and variable D_{12} (see (1)) become interactive. So, for this reason we cannot obtain the sets of possible values of the latest starting times of activity (1, 2) by calculating the difference between the set of possible values of the latest moment when event 2 may occur and of duration D_{12} (see (3)). Similarly for the set of possible values of floats of the activity (see (4)). We get the sets of possible values of latest starting times and floats of the activity equal to $[-1, 1]$. Thus the results are far from the expected $[0, 0]$.

Several authors tried to cope with this problem (see [3] for a brief survey and a full bibliography). Kaufmann and Gupta [15], Hapke et al. [14] and Rommelfanger [21] suggest substitutes to the interval (fuzzy) subtraction, so as to improve the situation, but these techniques remain ad hoc. Nasution [19] resorts to symbolic computations on the variable duration times. However, this technique is unwieldy and highly combinatorial.

In Sections 2.2 and 2.3, we give $\mathcal{O}(mn)$ algorithms, which correctly determine the sets of possible values of the latest starting times of an activity.

2.2. Determination of the minimal latest starting time of an activity

We first describe a problem closely related to PLBLST, that is the problem of evaluating *possible criticality* of an activity. We provide a solution for it, in some special case, that allows us to construct a polynomial algorithm for solving PLBLST.

2.2.1. Problem of evaluating the possible criticality of an activity—a polynomially solvable case

Let us recall the notions of the possible criticality of activities and paths in network G .

Definition 1. An activity $(k, l) \in A$ (resp. a path $p \in P$) is possibly critical in G if and only if there exists a configuration of times $\Omega \in \mathfrak{D}$ such that (k, l) (resp. p) is critical in G in the usual sense for Ω .

The possible criticality have been thoroughly investigated in [5,6]. The problem of the possible criticality for a path is polynomially solvable. Unfortunately, the same problem for an activity turns 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 [7]). In [11] a polynomial algorithm has been provided only in case of series–parallel networks.

The following proposition is obvious:

Proposition 1. An activity $(k, l) \in A$ is possibly critical in G if and only if it belongs to some possibly critical path $p \in P$.

Lemma 1 gives necessary and sufficient conditions for establishing the possible criticality of a given path $p \in P$.

Lemma 1 (Chanas and Zieliński [6]). *A path $p \in P$ is possibly critical in G if and only if p is critical in the usual sense for configuration $\Omega \in \mathfrak{D}$ determined in the following way:*

$$d_{ij}(\Omega) = \begin{cases} \bar{d}_{ij} & \text{for } (i, j) \in p, \\ \underline{d}_{ij} & \text{for } (i, j) \notin p. \end{cases}$$

Let us focus on a network G , in which some duration intervals are precisely given. It is worth noticing that in the problem of evaluating possible criticality of activity (k, l) durations of activities $(i, j) \in A \setminus PREC(k, l) \setminus SUCC(k, l) \setminus \{(k, l)\}$ may be always fixed at their lower bounds for all configurations of times. It follows directly from Definition 1, Propositions 6 and 7. If additionally activities $(i, j) \in PREC(k, l)$ have precise durations ($\underline{d}_{ij} = \bar{d}_{ij}$), then we can show that in this case the considered problem is polynomially solvable. For such a network, we define the set of possible configurations of activity durations and denote it by $\mathfrak{D}_{kl}^s(x)$, $x \geq 0$. $\mathfrak{D}_{kl}^s(x)$ is the Cartesian product of time intervals D_{ij} , $(i, j) \in A$, given as follows:

$$D_{ij} = \begin{cases} [\underline{d}_{ij}, \bar{d}_{ij}] & \text{for } (i, j) \in SUCC(k, l), \\ [x, x] & \text{for } (i, j) = (k, l), \\ [\underline{d}_{ij}, \underline{d}_{ij}] & \text{otherwise.} \end{cases} \quad (5)$$

Similarly, we define the set of possible configurations of activity durations $\mathfrak{D}_{kl}^p(x)$ if activities $(i, j) \in SUCC(k, l)$ have precise durations. $\mathfrak{D}_{kl}^p(x)$ is the Cartesian product of D_{ij} , $(i, j) \in A$, given as follows:

$$D_{ij} = \begin{cases} [\underline{d}_{ij}, \bar{d}_{ij}] & \text{for } (i, j) \in PREC(k, l), \\ [x, x] & \text{for } (i, j) = (k, l), \\ [\underline{d}_{ij}, \underline{d}_{ij}] & \text{otherwise.} \end{cases} \quad (6)$$

We are thus led to the following obvious property.

Proposition 2. *Let activities $(i, j) \in PREC(k, l)$ (resp. $(i, j) \in SUCC(k, l)$) have precise durations. Activity (k, l) is possibly critical in G if and only if there exists a configuration $\Omega \in \mathfrak{D}_{kl}^s(\bar{d}_{kl})$ (resp. $\Omega \in \mathfrak{D}_{kl}^p(\bar{d}_{kl})$) such that (k, l) is critical in the usual sense for Ω in G .*

Hence the problem of evaluating the possible criticality of (k, l) in G , in which the durations of activities $(i, j) \in PREC(k, l)$ (resp. $(i, j) \in SUCC(k, l)$) are precisely given, boils down to the one in G , in which the durations of activities $(i, j) \in A \setminus SUCC(k, l)$ (resp. $(i, j) \in A \setminus PREC(k, l)$) are fixed (see (5), (6) and Proposition 2).

We now give a polynomial algorithm (Algorithm 1) for asserting whether an activity $(k, l) \in A$ is possibly critical in G , under the assumption that activities $(i, j) \in SUCC(k, l)$ have precise durations. The logic of Algorithm 1 is to find a configuration $\Omega \in \mathfrak{D}_{kl}^s(\bar{d}_{kl})$ in which (k, l) is critical in the usual sense. To find a configuration a node labeling is performed with convenient setting of the activity durations. A node $i \in V$ is labeled *true* if the longest path from node 1 to i uses activity (k, l) in a systematically constructed configuration Ω , and *false* otherwise. The duration of activity $(i, j) \in A$ is set to its upper

bound if node i is labeled *true*, and to its lower bound otherwise, except for (k, l) . In this case the duration of (k, l) is set to its upper bound. It is worth pointing out that the algorithm is crucial and it will be substantially used in a algorithm for solving PLBLST. We will illustrate it with an example in Section 2.2.2.

Algorithm 1 Asserting whether an activity is possibly critical

Require: A network $G = \langle V, A \rangle$, a specified activity $(k, l) \in A$, time intervals $D_{ij} = [\underline{d}_{ij}, \bar{d}_{ij}]$, $(i, j) \in A$.

Ensure: A configuration $\Omega \in \mathfrak{D}_{kl}^s(\bar{d}_{kl})$, $PossCritical = true$ if (k, l) is possibly critical, *false* otherwise.

```

▷ PHASE 1:
1:  $t_1^e \leftarrow 0$ ;  $label(1) \leftarrow false$ ;
2: for  $j \leftarrow 2$  to  $l - 1$  do
3:   for all  $i \in Prec(j)$  do
4:      $d_{ij} \leftarrow \underline{d}_{ij}$ 
5:   end for
6:    $t_j^e \leftarrow \max\{t_i^e + d_{ij} \mid i \in Prec(j)\}$ ;  $label(j) \leftarrow false$ 
7: end for
8: for all  $i \in Prec(l)$  do
9:   if  $i \neq k$  then
10:     $d_{il} \leftarrow \underline{d}_{il}$ 
11:   end if
12: end for
13:  $d_{kl} \leftarrow \bar{d}_{kl}$ ;
14:  $t_l^e \leftarrow \max\{t_i^e + d_{il} \mid i \in Prec(l)\}$ ;  $label(l) \leftarrow false$ ;
15: if  $t_l^e \neq t_k^e + d_{kl}$  then
16:    $PossCritical \leftarrow false$ ; exit
17: end if
▷ PHASE 2:
18:  $label(l) \leftarrow true$ ;
19: for  $j \leftarrow l + 1$  to  $n$  do
20:   for all  $i \in Prec(j)$  do
21:     if  $label(i) = true$  then
22:        $d_{ij} \leftarrow \bar{d}_{ij}$ 
23:     else
24:        $d_{ij} \leftarrow \underline{d}_{ij}$ 
25:     end if
26:   end for
27:    $t_j^e \leftarrow \max\{t_i^e + d_{ij} \mid i \in Prec(j)\}$ ;
28:   if  $\{i \mid i \in Prec(j), label(i) = true, t_j^e = t_i^e + d_{ij}\} \neq \emptyset$  then
29:      $label(j) \leftarrow true$ 
30:   else
31:      $label(j) \leftarrow false$ 
32:   end if
33: end for
34:  $PossCritical \leftarrow label(n)$ ;

```

The correctness Algorithm 1 follows directly from Lemma 2.

Lemma 2. *Let activities $(i, j) \in PREC(k, l)$ have precise durations. Activity (k, l) is possibly critical in network G if and only if Algorithm 1 returns $PossCritical = true$ for (k, l) .*

Proof. (\implies) Assume that the activity (k, l) is possibly critical in G in which activities $(i, j) \in PREC(k, l)$ have precise durations. Then there exists a possibly critical path $p^* \in P(k, l)$ (see Proposition 1). Hence the path p^* is critical in the usual sense in G in the configuration $\Omega' \in \mathfrak{D}_{kl}^s(\bar{d}_{kl})$ determined as in Lemma 1 (see Proposition 2). Consequently the length of p^* fulfills the following inequality:

$$l_{p^*}(\Omega') \geq l_p(\Omega') \quad \text{for all } p \notin P(k, l). \tag{7}$$

We claim that Algorithm 1 returns $PossCritical = true$ for (k, l) . To prove this, assume on the contrary that Algorithm 1 returns $PossCritical = false$. Note that it also returns $label(i), i = 1, \dots, n$. Let $\Omega'' \in \mathfrak{D}_{kl}^s(\bar{d}_{kl})$ be a configuration of activity durations after the termination of Algorithm 1. $PossCritical = false$, and so either $label(l) = false$ or $label(n) = false$.

Consider the case when $label(l) = false$. Then, (k, l) does not lie on any critical path in configuration Ω'' , $t_l^e(\Omega'') \neq t_k^e(\Omega'') + d_{kl}(\Omega'')$ (see Algorithm 1, rows: 15, 16). From this and the fact that $d_{ij}(\Omega') = d_{ij}(\Omega'')$ for $(i, j) \notin SUCC(k, l)$, $\Omega', \Omega'' \in \mathfrak{D}_{kl}^s(\bar{d}_{kl})$, it follows that there exists a path in Ω' , which is longer than p^* , contrary to (7).

Consider the case when $label(n) = false$. Then, $label(l) = true$. Let us choose the subpath of p^* from k to r , we denote it by p_{kr}^* , whose nodes $i \in p_{kr}^*$ are labeled $true$ except for start node k and end node r ($label(k) = false$ and $label(r) = false$). Such subpath exists, since $p^* \in P$ and $label(n) = false$. Hence durations $d_{ij}(\Omega'')$ of activities $(i, j) \in p_{kr}^*$ are equal to \bar{d}_{ij} (see Algorithm 1, rows: 13, 22). The result is $d_{ij}(\Omega'') = d_{ij}(\Omega')$ for $(i, j) \in p_{kr}^*$. This equality also holds for $(i, j) \notin SUCC(k, l)$. Since $label(r) = false$, there exists a path p'_{1r} from 1 to r that does not use (k, l) , with activities durations in Ω'' at their lower bounds and of a length longer than the length of p'_{1r} in Ω'' (see Algorithm 1, rows: 27, 28, 31). Let us concatenate p'_{1r} with the subpath of p^* from r to n . Then we obtain the path $p' \in P$, $p' \notin P(k, l)$, with the length $l_{p'}(\Omega'')$ such that $l_{p^*}(\Omega'') < l_{p'}(\Omega'')$. It is clear that this inequality holds in Ω' , contrary to (7). Thus we arrive to contradiction, since p^* traversing (k, l) is not critical in Ω' .

(\impliedby) Assume that Algorithm 1 returns $PossCritical = true$ for (k, l) and a configuration $\Omega \in \mathfrak{D}_{kl}^s(\bar{d}_{kl})$. Then, $label(l) = true$ and $label(n) = true$. Hence, there exist a longest path from 1 to l that uses (k, l) and a path from l to n whose nodes are labeled $true$ in Ω (see Algorithm 1, rows: 27, 28, 29). These paths form a critical path containing (k, l) which is, by Definition 1, possibly critical one. By Proposition 1, (k, l) is possibly critical. \square

It is clear that the running time of Algorithm 1 is $\mathcal{O}(m)$.

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

2.2.2. A polynomial algorithm for determining the lower bound on the latest starting time of an activity

We now present an algorithm for solving PLBLST. Let us recall an important result, given by Dubois et al. [9], that allows to reduce the set of configurations \mathfrak{D} .

Proposition 3. $t_{-kl}^l = \min_{\Omega \in \mathfrak{D}_{kl}^s(\bar{d}_{kl})} t_{kl}^l(\Omega)$. Moreover, the minimum $t_{kl}^l(\Omega)$ is attained on the vertices of the hyper-rectangle $\mathfrak{D}_{kl}^s(\bar{d}_{kl})$.

The key to constructing the algorithm for computing t_{kl}^l is Lemma 3. The idea of the algorithm (Algorithm 2) is based on this lemma. It consists in finding the minimal nonnegative real number f_{kl}^* that added to the upper bound of duration interval of a specified (k, l) makes it possibly critical.

Lemma 3. *Let f_{kl}^* be the minimal nonnegative real number such that (k, l) with a duration $\bar{d}_{kl} + f_{kl}^*$ becomes possibly critical. Then $t_k^e + f_{kl}^* = \min_{\Omega \in \mathfrak{D}_{kl}^s(\bar{d}_{kl})} t_{kl}^l(\Omega)$, where t_k^e is the earliest moment when event k occurs.*

Proof. Let us observe that $t_{kl}^l(\Omega) = t_k^e + f_{kl}(\Omega)$ for all $\Omega \in \mathfrak{D}_{kl}^s(\bar{d}_{kl})$. This follows from the fact that $t_k^e(\Omega)$ is equal to t_k^e for all $\Omega \in \mathfrak{D}_{kl}^s(\bar{d}_{kl})$. Thus to prove $t_k^e + f_{kl}^* = \min_{\Omega \in \mathfrak{D}_{kl}^s(\bar{d}_{kl})} t_{kl}^l(\Omega)$, we only need to show that $f_{kl}^* = \min_{\Omega \in \mathfrak{D}_{kl}^s(\bar{d}_{kl})} f_{kl}(\Omega)$. Assume on the contrary that $f_{kl}^* \neq \min_{\Omega \in \mathfrak{D}_{kl}^s(\bar{d}_{kl})} f_{kl}(\Omega)$. Let us consider two cases.

Case 1: $f_{kl}^* < \min_{\Omega \in \mathfrak{D}_{kl}^s(\bar{d}_{kl})} f_{kl}(\Omega)$. The result is $\min_{\Omega \in \mathfrak{D}_{kl}^s(\bar{d}_{kl})} f_{kl}(\Omega) > 0$ and consequently (k, l) with duration $\bar{d}_{kl} + f_{kl}^*$ is not critical for all $\Omega \in \mathfrak{D}_{kl}^s(\bar{d}_{kl} + f_{kl}^*)$. This contradicts the possible criticality of (k, l) .

Case 2: $f_{kl}^* > \min_{\Omega \in \mathfrak{D}_{kl}^s(\bar{d}_{kl})} f_{kl}(\Omega)$. Hence there exists a configuration $\Omega' \in \mathfrak{D}_{kl}^s(\bar{d}_{kl})$ such that $f_{kl}^* > f_{kl}(\Omega')$. Let us increase the duration of (k, l) from \bar{d}_{kl} to $\bar{d}_{kl} + f_{kl}(\Omega')$ in Ω' . For this new configuration, say Ω'' , $\Omega'' \in \mathfrak{D}_{kl}^s(\bar{d}_{kl} + f_{kl}(\Omega'))$, (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 $\bar{d}_{kl} + f_{kl}^*$ becomes possibly critical. \square

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 $D_{ij} = [d_{ij}, \bar{d}_{ij}]$, $(i, j) \in A$

Ensure: The minimal latest starting time of (k, l) , t_{kl}^l .

```

1:  $f_{kl} \leftarrow 0$ ;
    $\triangleright$  Check possible criticality of  $(k, l)$ .
2: call Algorithm 1;
3: while  $PossCritical = false$  do
4:   if  $label(l) = true$  then
5:      $\Delta \leftarrow \min\{t_j^e - t_i^e - d_{ij} \mid (i, j) \in A, label(i) = true, label(j) = false\}$ 
6:   else
7:      $\Delta \leftarrow t_l^e - t_k^e - d_{kl}$ 
8:   end if
9:    $t_l^e \leftarrow t_l^e + \Delta$ ;
10:   $f_{kl} \leftarrow f_{kl} + \Delta$ ;
    $\triangleright$  Check possible criticality of  $(k, l)$  with implicitly
    $\triangleright$  increased duration  $\bar{d}_{kl} + f_{kl}$ .
11: call only PHASE 2 of Algorithm 1
12: end while
13:  $t_{kl}^l \leftarrow t_k^e + f_{kl}$ ;  $\triangleright f_{kl}$  equals  $f_{kl}^*$ 

```

In each iteration of Algorithm 2 the duration of (k, l) is suitably augmented (row 9) and the possible criticality of (k, l) is tested. The testing boils down to applying Algorithm 1 (rows: 2, 11). This process is repeated until (k, l) becomes possibly critical. Then from Lemma 3, we immediately obtain its minimal

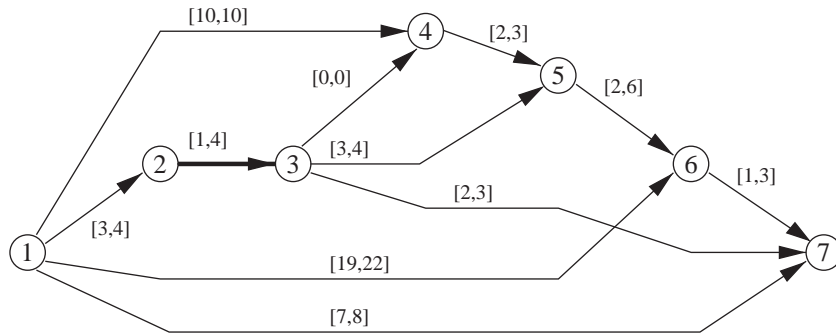


Fig. 1. A network with duration intervals.

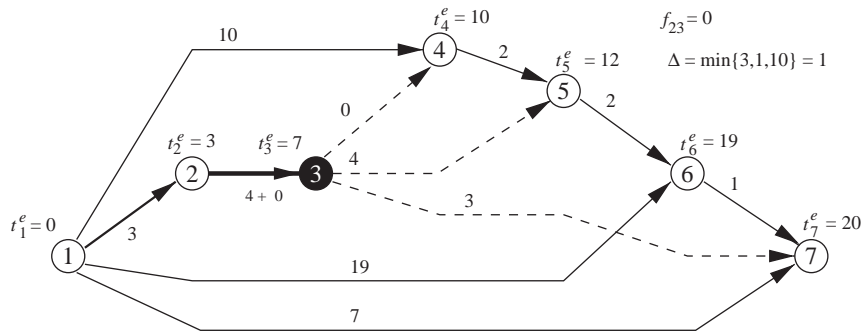


Fig. 2. The network, given in Fig. 1, after the first call of Algorithm 1 in Algorithm 2.

latest starting time. Note that, Algorithm 1 returns either a configuration of durations in which (k, l) is critical or a configuration that makes (k, l) as close as possible to being critical, stored in variables d_{ij} , $(i, j) \in A$. It also returns node labels $label(i)$ and the earliest moments of the occurrence of events t_i^e , $i = 1, \dots, n$, in these configurations. If $PossCritical = false$, i.e. either $label(l) = false$ or $label(n) = false$, then duration d_{kl} (or equivalently the earliest moment when event l occurs t_l^e), in the obtained configuration, is augmented by quantity Δ (row 9) being the minimum over free floats of activities whose the start and end node are labeled *true* and *false*, respectively (rows 5 or 7). This means that there exist the longest paths traversing (k, l) from node 1 to the start nodes of all these activities in the returned configuration. After increasing the duration of (k, l) in the next call of Algorithm 1 at least one of the end nodes of the activities will be labeled *true*. The aim is to label *true* node n (to construct a configuration in which (k, l) will be critical) in consecutive calls. It is worth noticing that in row 11 only PHASE 2 of Algorithm 1 is called, because the earliest moments of the occurrence of events t_i^e , node labels $label(i)$, and durations d_{ij} , for $i, j \leq l$, computed in the first call (row 2) remain unchanged.

To clarify Algorithm 2, we apply it to the network in Fig. 1. Activity $(2, 3)$ is distinguished. The network after the first, second and third call of Algorithm 1 is given in Figs. 2, 3 and 4, respectively. Nodes labeled *true* after the calls of Algorithm 1 are black, activities whose free floats are computed in row 5 (see Algorithm 2) are marked with a dashed line, durations of activities that are at their upper bounds are in

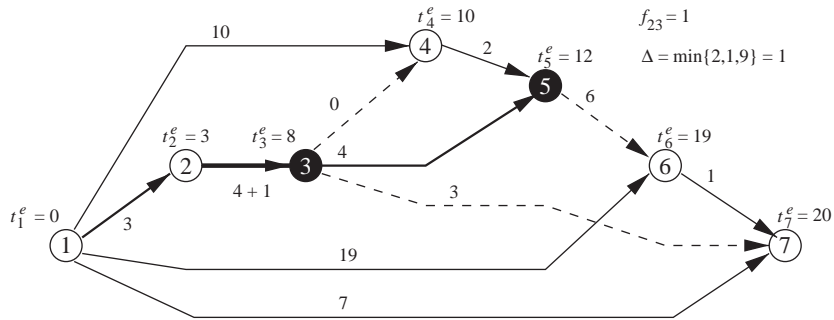


Fig. 3. The network, given in Fig. 1, after the second call of Algorithm 1 in Algorithm 2.

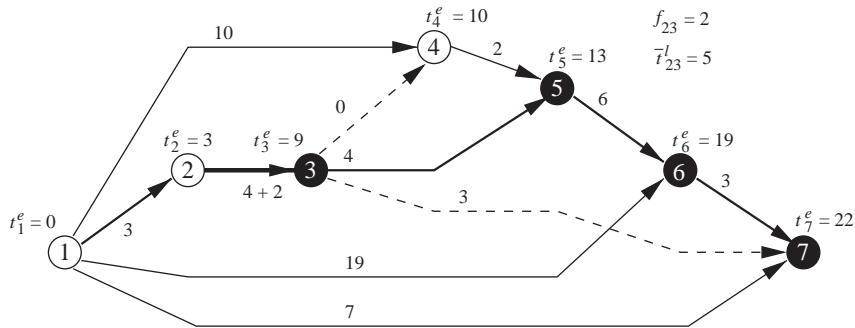


Fig. 4. The network, given in Fig. 1, after the third call of Algorithm 1—PHASE 2—in Algorithm 2.

bold print. A possibly critical path, determined by Algorithm 1, that traverses activity (2, 3) is presented in Fig. 4, $1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 6 \rightarrow 7$. Thus (2, 3) is possibly critical. The number $f_{23} = 2$, in Fig. 4, is the minimal one such that when added to \bar{d}_{23} , it makes (2, 3) possibly critical. From this and Lemma 3, we obtain the minimal latest starting time $t_{23}^l = 5$.

Lemma 4. Algorithm 2 finds t_{kl}^l .

Proof. To prove that a quantity $t_k^e + f_{kl}$, computed by Algorithm 2 (row 13), is the minimal latest starting time of activity (k, l), we only need to show, by Lemma 3, that f_{kl} is the minimal nonnegative real number such that (k, l) with duration equal to $\bar{d}_{kl} + f_{kl}$ becomes possibly critical.

We use a proof by contradiction. Suppose that there exists a nonnegative real number f'_{kl} such that (k, l) becomes possibly critical for a duration equals $\bar{d}_{kl} + f'_{kl}$ and

$$f'_{kl} < f_{kl}. \tag{8}$$

Let us denote by q the number of calls of Algorithm 1 in Algorithm 2. This yields $f_{kl} = \sum_{i=1}^{q-1} \Delta_i$ (see Algorithm 2, row 10). Note that for the duration of (k, l) equal to $\bar{d}_{kl} + \sum_{i=1}^{q-2} \Delta_i$, Algorithm 1

asserts that (k, l) is not possibly critical. Additionally, it returns a configuration of durations $\Omega', \Omega' \in \mathfrak{D}_{kl}^s(\bar{d}_{kl} + \sum_{i=1}^{q-2} \Delta_i)$, node labels $label_i(\Omega')$, $i = 1, \dots, n$ ($label_i(\Omega')$ temporary stands for the label of i in configuration Ω'). For the duration of (k, l) equal to $\bar{d}_{kl} + f'_{kl}$, Algorithm 1 asserts the possible criticality of (k, l) and returns a configuration of durations $\Omega'', \Omega'' \in \mathfrak{D}_{kl}^s(\bar{d}_{kl} + f'_{kl})$, node labels $label_i(\Omega'')$, $i = 1, \dots, n$. Since (k, l) is possibly critical for $\bar{d}_{kl} + f'_{kl}$, but for $\bar{d}_{kl} + \sum_{i=1}^{q-2} \Delta_i$ (k, l) is not, we have $f'_{kl} > \sum_{i=1}^{q-2} \Delta_i$. Consequently $f'_{kl} = \sum_{i=1}^{q-2} \Delta_i + \delta$ ($\delta > 0$). Moreover, from the way of determining configurations Ω' and Ω'' by Algorithm 1 for $\bar{d}_{kl} + \sum_{i=1}^{q-2} \Delta_i$ and $\bar{d}_{kl} + \sum_{i=1}^{q-2} \Delta_i + \delta$, respectively, we deduce that there exists at least one activity $(i', j') \in A$ such that

$$\begin{aligned} label_{i'}(\Omega') &= true, label_{j'}(\Omega') = false, \\ label_{i'}(\Omega'') &= true, label_{j'}(\Omega'') = true \end{aligned}$$

or

$$label_{i'}(\Omega') = false, label_{j'}(\Omega'') = true \text{ for } (i', j') = (k, l)$$

with free float

$$t_{j'}^e(\Omega') - t_{i'}^e(\Omega') - d_{i',j'}(\Omega') \leq \delta. \tag{9}$$

We now focus on the stage after $(q - 2)$ th call of Algorithm 1. Algorithm 2 in rows 5 or 7 determines $\Delta_{q-1} = \min\{t_{j'}^e(\Omega') - t_{i'}^e(\Omega') - d_{i',j'}(\Omega') \mid (i, j) \neq (k, l), label_i(\Omega') = true, label_j(\Omega') = false \text{ or } (i, j) = (k, l), label_j(\Omega') = false\}$. Combining this and (9), we conclude $\Delta_{q-1} \leq \delta$. Hence $\sum_{i=1}^{q-2} \Delta_i + \Delta_{q-1} \leq \sum_{i=1}^{q-2} \Delta_i + \delta$ and therefore $f_{kl} \leq f'_{kl}$, contrary to (8). \square

Lemma 5. *The running time of Algorithm 2 is $\mathcal{O}(mn)$.*

Proof. In each call of Algorithm 1 at least one node is labeled *true* that in previous calls was labeled *false* (nodes labeled *true* in previous calls have unchanged labels in consecutive calls). Thus, Algorithm 1 ($\mathcal{O}(m)$) is invoked, in Algorithm 2, $\mathcal{O}(n)$ times (until $label(n) = true$). Hence the entire complexity of Algorithm 2 is $\mathcal{O}(mn)$. \square

2.3. Determination of the maximal latest starting time of an activity

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

2.3.1. Problem of evaluating the necessary criticality of an activity—a polynomially solvable case

Let us recall the notions of the necessary criticality of activities and paths in the network G .

Definition 2. An activity $(k, l) \in A$ (resp. a path $p \in P$) is necessarily critical in G if and only if for every configuration of times $\Omega \in \mathfrak{D}$, (k, l) (resp. p) is critical in G in the usual sense.

The notions of the necessary criticality, both with respect to paths and activities, have been analyzed in [3]. 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 [11] a polynomial algorithm has been provided only for series–parallel networks.

Let us consider a polynomially solvable case of the problem of evaluating the necessary criticality of distinguished activity (k, l) . From Definition 2, Propositions 6 and 7 follow that durations of activities $(i, j) \in A \setminus PREC(k, l) \setminus SUCC(k, l) \setminus \{(k, l)\}$ may be always fixed at their upper bounds for all configurations of times in network G . If we additionally assume that activities $(i, j) \in PREC(k, l)$ in G have precise durations ($\underline{d}_{ij} = \bar{d}_{ij}$), then we can show that in this special case the problem of asserting necessary criticality of (k, l) is polynomially solvable. For such a network, we define the set of possible configurations of activity durations and denote it by $\overline{\mathfrak{D}}_{kl}^s(x)$, $x \geq 0$. $\overline{\mathfrak{D}}_{kl}^s(x)$ is the Cartesian product of D_{ij} , $(i, j) \in A$, given as follows:

$$D_{ij} = \begin{cases} [\underline{d}_{ij}, \bar{d}_{ij}] & \text{for } (i, j) \in SUCC(k, l), \\ [x, x] & \text{for } (i, j) = (k, l), \\ [\bar{d}_{ij}, \bar{d}_{ij}] & \text{otherwise.} \end{cases} \quad (10)$$

In the same manner, we can define the set of possible configurations of activity durations $\overline{\mathfrak{D}}_{kl}^p(x)$, $x \geq 0$, if activities $(i, j) \in SUCC(k, l)$ have precise durations. $\overline{\mathfrak{D}}_{kl}^p(x)$ is the Cartesian product of D_{ij} , $(i, j) \in A$, given as follows:

$$D_{ij} = \begin{cases} [\underline{d}_{ij}, \bar{d}_{ij}] & \text{for } (i, j) \in PREC(k, l), \\ [x, x] & \text{for } (i, j) = (k, l), \\ [\bar{d}_{ij}, \bar{d}_{ij}] & \text{otherwise.} \end{cases} \quad (11)$$

We have thus proved the following property:

Proposition 4. *Let activities $(i, j) \in PREC(k, l)$ (resp. $(i, j) \in SUCC(k, l)$) have precise durations. Activity (k, l) is not necessarily critical in G if and only if there exists a configuration $\Omega \in \overline{\mathfrak{D}}_{kl}^s(\underline{d}_{kl})$ (resp. $\Omega \in \overline{\mathfrak{D}}_{kl}^p(\underline{d}_{kl})$) such that none of critical paths, in Ω , uses (k, l) .*

Thus, the problem of evaluating the necessary criticality of (k, l) in G , in which the durations of activities $(i, j) \in PREC(k, l)$ (resp. $(i, j) \in SUCC(k, l)$) are precisely given, boils down to the one in G , in which the durations of activities $(i, j) \in A \setminus SUCC(k, l)$ (resp. $(i, j) \in A \setminus PREC(k, l)$) are fixed (see Proposition 4 and (10), (11)).

Algorithm 3 enables us to assert whether an activity (k, l) is necessarily critical in G , under the assumption that activities $(i, j) \in SUCC(k, l)$ have precise durations. The main idea of an algorithm (Algorithm 3) which can evaluate the necessary criticality of activity (k, l) is to find a configuration $\Omega \in \overline{\mathfrak{D}}_{kl}^s(\underline{d}_{kl})$ in which (k, l) is not critical in the usual sense. If such a configuration Ω is successfully determined then (k, l) is not necessarily critical, otherwise it is. The algorithm is similar in spirit to Algorithm 1. The approach here applied is complementary to the one in Algorithm 1. To find a configuration Ω , node labeling is performed with a convenient setting of the activity durations. A node $i \in V$ is labeled *true* if the longest path from node 1 to i uses activities parallel to (k, l) , and it is longer than all paths from 1 to i traversing (k, l) , in a systematically constructed configuration Ω , and *false* otherwise. The duration of activity $(i, j) \in A$ is set to its upper bound if node i is labeled *true*, and to its lower bound

otherwise, except for (k, l) . In this case the duration of (k, l) is set to its lower bound. We will illustrate the algorithm with an example in Section 2.3.2.

Algorithm 3 Asserting whether an activity is necessarily critical

Require: A network $G = \langle V, A \rangle$, a specified activity $(k, l) \in A$, time intervals $D_{ij} = [\underline{d}_{ij}, \bar{d}_{ij}]$, $(i, j) \in A$.

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

▷ PHASE 1:

```

1:  $t_1^e \leftarrow 0$ ;  $label(1) \leftarrow true$ ;
2: for  $j \leftarrow 2$  to  $l - 1$  do
3:   for all  $i \in Prec(j)$  do
4:      $d_{ij} \leftarrow \bar{d}_{ij}$ 
5:   end for
6:    $t_j^e \leftarrow \max\{t_i^e + d_{ij} \mid i \in Prec(j)\}$ ;  $label(j) \leftarrow true$ 
7: end for
8: for all  $i \in Prec(l)$  do
9:   if  $i \neq k$  then
10:     $d_{il} \leftarrow \bar{d}_{il}$ 
11:   end if
12: end for
13:  $d_{kl} \leftarrow \underline{d}_{kl}$ ;
14:  $t_l^e \leftarrow \max\{t_i^e + d_{il} \mid i \in Prec(l)\}$ ;  $label(l) \leftarrow true$ ;
15: if  $t_l^e \neq t_k^e + d_{kl}$  then
16:    $NecCritical \leftarrow false$ ; exit
17: end if
▷ PHASE 2:
18:  $label(l) \leftarrow false$ ;
19: for  $j \leftarrow l + 1$  to  $n$  do
20:   for all  $i \in Prec(j)$  do
21:     if  $label(i) = true$  then
22:        $d_{ij} \leftarrow \bar{d}_{ij}$ 
23:     else
24:        $d_{ij} \leftarrow \underline{d}_{ij}$ 
25:     end if
26:   end for
27:    $t_j^e \leftarrow \max\{t_i^e + d_{ij} \mid i \in Prec(j)\}$ ;
28:   if  $\{i \mid i \in Prec(j), label(i) = false, t_j^e = t_i^e + d_{ij}\} \neq \emptyset$  then
29:      $label(j) \leftarrow false$ 
30:   else
31:      $label(j) \leftarrow true$ 
32:   end if
33: end for
34:  $NecCritical \leftarrow \text{not } label(n)$ ;

```

The correctness of Algorithm 3 follows from the following lemma:

Lemma 6. *Let activities $(i, j) \in PREC(k, l)$ have precise durations. Activity (k, l) is not necessarily critical in G if and only if Algorithm 3 returns $NecCritical = false$ for (k, l) .*

Proof. (\implies) Suppose the assertion of the lemma is false, i.e. $NecCritical = true$. We denote the configuration of the activity durations after the termination by $\Omega' \in \overline{\mathfrak{D}}_{kl}^s(\underline{d}_{kl})$. By assumption, activity (k, l) is not necessarily critical in G in which activities $(i, j) \in PREC(k, l)$ have precise durations, hence there exists a configuration $\Omega^* \in \overline{\mathfrak{D}}_{kl}^s(\underline{d}_{kl})$ such that none of the critical paths uses (k, l) in Ω^* (see Proposition 4). Consequently there exists a critical path p^* , that traverses activities parallel to (k, l) , $p^* \notin P(k, l)$, with the length $l_{p^*}(\Omega^*)$ satisfying the following inequality

$$l_{p^*}(\Omega^*) > l_p(\Omega^*) \quad \text{for all } p \in P(k, l). \quad (12)$$

Moreover, p^* is possibly critical and therefore it is critical in the configuration $\Omega'' \in \overline{\mathfrak{D}}_{kl}^s(\underline{d}_{kl})$ determined as in Lemma 1. It easy to check that inequality (12) holds in Ω'' . (12) now becomes

$$l_{p^*}(\Omega'') > l_p(\Omega'') \quad \text{for all } p \in P(k, l). \quad (13)$$

Note that (13) states that (k, l) is not necessarily critical.

Let us choose a subpath of p^* from 1 to r , we denote it by p_{1r}^* , whose nodes $i \in p_{1r}^*$ are labeled *true* except for end node r . p_{1r}^* exists, since $p \in P$ and $NecCritical = true$ and in consequence $label(n) = false$ (see Algorithm 3, row 34). Hence durations $d_{ij}(\Omega')$ of activities $(i, j) \in p_{1r}^*$ are equal to their upper bounds (see Algorithm 3, rows: 4, 22). This forces $d_{ij}(\Omega') = d_{ij}(\Omega'')$ for all $(i, j) \in p_{1r}^*$. The same equality holds for all $(i, j) \notin SUCC(k, l)$, since $\Omega', \Omega'' \in \overline{\mathfrak{D}}_{kl}^s(\underline{d}_{kl})$. Since $label(r) = false$, there exists a path p'_{1r} from 1 to r that uses (k, l) with durations in Ω' at their lower bounds and of the length greater or equal to the length p_{1r}^* in Ω' (see Algorithm 3, rows: 27, 28, 29). If we now concatenate p'_{1r} with the subpath of p^* from r to n then we obtain the path $p' \in P(k, l)$ with the length $l_{p'}(\Omega')$ such that $l_{p^*}(\Omega') \leq l_{p'}(\Omega')$. It is easily seen that this inequality holds for configuration Ω'' , contrary to (13).

(\impliedby) Assume that Algorithm 3 returns $NecCritical = false$ and a configuration $\Omega^* \in \overline{\mathfrak{D}}_{kl}^s(\underline{d}_{kl})$. Then $label(l) = true$ or $label(n) = true$. From this, it may be concluded that none of the critical paths uses (k, l) in Ω^* (see Algorithm 3, rows: 14, 15, 16, 27, 28, 31). By Proposition 4, (k, l) is not necessarily critical. This completes the proof. \square

It is evident that the running time of Algorithm 3 is $\mathcal{O}(m)$.

Remark 2. The algorithm asserting whether an activity (k, l) is necessarily critical in G with activities $(i, j) \in SUCC(k, l)$ having precise durations can be identical to Algorithm 3. It is sufficient to reverse arcs in network G and carry out the computations from node n down to 1.

2.3.2. A polynomial algorithm for determining the upper bound on the latest starting time of an activity

Here, we propose an algorithm for solving PUBLST. Let us recall an important result, given by Dubois et al. [9], that allows to reduce the set of configurations \mathfrak{D} .

Proposition 5. $\bar{t}_{kl}^l = \max_{\Omega \in \overline{\mathfrak{D}}_{kl}^s(\underline{d}_{kl})} t_{kl}^l(\Omega)$. Moreover, the maximum $t_{kl}^l(\Omega)$ is attained on the vertices of the hyper-rectangle $\overline{\mathfrak{D}}_{kl}^s(\underline{d}_{kl})$.

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 $D_{ij} = [\underline{d}_{ij}, \bar{d}_{ij}]$, $(i, j) \in A$

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

- 1: $f_{kl} \leftarrow 0$;
 \triangleright Check necessary criticality of (k, l) .
 - 2: call Algorithm 3;
 - 3: **while** $NecCritical = false$ **do**
 - 4: **if** $label(l) = true$ **then**
 - 5: $\Delta \leftarrow t_l^e - t_k^e - d_{kl}$
 - 6: **else**
 - 7: $\Delta \leftarrow \min\{t_j^e - t_i^e - d_{ij} \mid (i, j) \in A, label(i) = false, label(j) = true\}$
 - 8: **end if**
 - 9: $t_l^e \leftarrow t_l^e + \Delta$;
 - 10: $f_{kl} \leftarrow f_{kl} + \Delta$;
 \triangleright Check necessary criticality of (k, l) with implicitly
 \triangleright increased duration $\underline{d}_{kl} + f_{kl}$.
 - 11: call only PHASE 2 of Algorithm 3
 - 12: **end while**
 - 13: $\bar{t}_{kl}^l \leftarrow t_k^e + f_{kl}$; $\triangleright f_{kl}$ equals f_{kl}^*
-

The main idea of the algorithm for determining \bar{t}_{kl}^l of a given activity $(k, l) \in A$ (Algorithm 4) is based on Lemma 7. It consists in determining the minimal nonnegative real number f_{kl}^* that added to the lower bound of the duration interval of a specified activity (k, l) makes it necessarily critical.

Lemma 7. *Let f_{kl}^* be the minimal nonnegative real number such that (k, l) with a duration $\underline{d}_{kl} + f_{kl}^*$ becomes necessarily critical. Then $\bar{t}_k^e + f_{kl}^* = \max_{\Omega \in \bar{\mathfrak{D}}_{kl}^s(\underline{d}_{kl})} t_{kl}^l(\Omega)$, where \bar{t}_k^e is the earliest moment when event k occurs.*

Proof. Our proof starts with the observation that $t_{kl}^l(\Omega) = \bar{t}_k^e + f_{kl}(\Omega)$, for all $\Omega \in \bar{\mathfrak{D}}_{kl}^s(\underline{d}_{kl})$. The observation follows from the fact that $t_k^e(\Omega) = \bar{t}_k^e$, for all $\Omega \in \bar{\mathfrak{D}}_{kl}^s(\underline{d}_{kl})$. Hence in order to prove $\bar{t}_k^e + f_{kl}^* = \max_{\Omega \in \bar{\mathfrak{D}}_{kl}^s(\underline{d}_{kl})} t_{kl}^l(\Omega)$, it suffices to show that $f_{kl}^* = \max_{\Omega \in \bar{\mathfrak{D}}_{kl}^s(\underline{d}_{kl})} f_{kl}(\Omega)$. Suppose on the contrary that $f_{kl}^* \neq \max_{\Omega \in \bar{\mathfrak{D}}_{kl}^s(\underline{d}_{kl})} f_{kl}(\Omega)$. Then we should consider the following two cases.

Case 1: $f_{kl}^* < \max_{\Omega \in \bar{\mathfrak{D}}_{kl}^s(\underline{d}_{kl})} f_{kl}(\Omega)$. This implies that there exists a configuration $\Omega' \in \bar{\mathfrak{D}}_{kl}^s(\underline{d}_{kl})$ such that $f_{kl}^* < f_{kl}(\Omega')$, which gives $f_{kl}(\Omega') > 0$. Consequently (k, l) is not critical in Ω' . Let us increase the duration of (k, l) from \underline{d}_{kl} to $\underline{d}_{kl} + f_{kl}^*$ in Ω' . For this new configuration, say Ω'' , $\Omega'' \in \bar{\mathfrak{D}}_{kl}^s(\underline{d}_{kl} + f_{kl}^*)$, (k, l) is still not critical, which contradicts the assumption that (k, l) is critical for all $\Omega \in \bar{\mathfrak{D}}_{kl}^s(\underline{d}_{kl} + f_{kl}^*)$.

Case 2: $f_{kl}^* > \max_{\Omega \in \bar{\mathfrak{D}}_{kl}^s(\underline{d}_{kl})} f_{kl}(\Omega)$. Thus (k, l) is critical with duration $\underline{d}_{kl} + \max_{\Omega \in \bar{\mathfrak{D}}_{kl}^s(\underline{d}_{kl})} f_{kl}(\Omega)$, for all $\Omega \in \bar{\mathfrak{D}}_{kl}^s(\underline{d}_{kl} + \max_{\Omega \in \bar{\mathfrak{D}}_{kl}^s(\underline{d}_{kl})} f_{kl}(\Omega))$, and therefore it is necessarily critical. This contradicts the assumption that f_{kl}^* is the minimal number such that (k, l) becomes necessarily critical. \square

In each iteration of Algorithm 4, the duration of (k, l) is suitably increased (row 9) and its necessary criticality is evaluated. The evaluation boils down to applying Algorithm 3. This process is repeated until (k, l) becomes necessarily critical. Then Lemma 7 gives the maximal latest starting time of (k, l) . Note

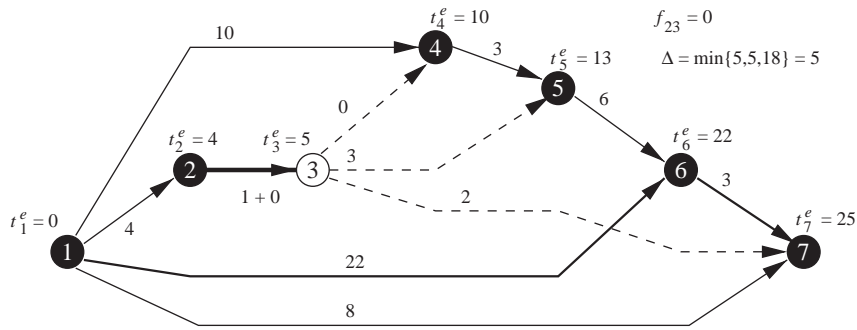


Fig. 5. The network, given in Fig. 1, after the first call of Algorithm 3 in Algorithm 4.

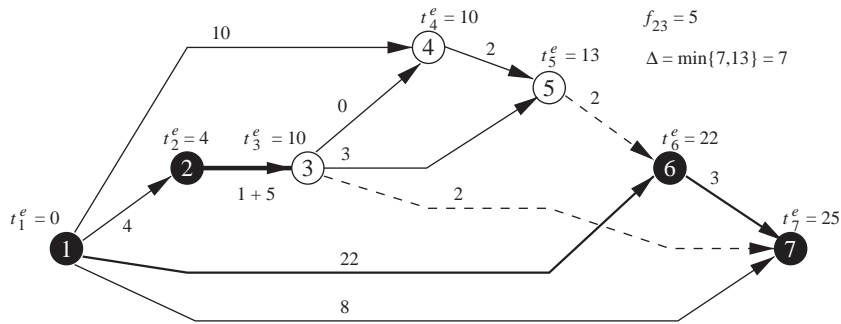


Fig. 6. The network, given in Fig. 1, after the second call of Algorithm 3—PHASE 2—in Algorithm 4.

that, Algorithm 3 tries to construct a configuration in which (k, l) is not critical, i.e. none of the longest paths uses (k, l) , but it may fail ($NecCritical = true$). In this case (k, l) is necessarily critical. Algorithm 3 returns the configuration of durations stored in variables d_{ij} , $(i, j) \in A$, node labels $label(i)$ and the earliest moments of the occurrence of events t_i^e , $i = 1, \dots, n$. If $NecCritical = false$, i.e. either $label(l) = true$ or $label(n) = true$, then duration d_{kl} (or equivalently the earliest moment when event l occurs t_l^e), in the returned configuration, is augmented by quantity Δ (row 9) being the minimum over free floats of activities whose the start and the end node are labeled *false* and *true*, respectively (rows 5 or 7). This means that none of the longest paths from node 1 to the end nodes of all these activities uses (k, l) in the obtained configuration. After augmenting the duration of (k, l) in the next call of Algorithm 3 at least one of the end nodes of the activities will be labeled *false*. The aim is to label *false* node n (to cause the failure of Algorithm 3 in constructing a configuration in which (k, l) is not critical) in consecutive calls. It is worth noticing that in row 11 only PHASE 2 of Algorithm 3 is called, because the earliest moments of the occurrence of events t_i^e , node labels $label(i)$, and durations d_{ij} , for $i, j \leq l$, computed in the first call (row 2) remain unchanged.

We now clarify Algorithm 4 by an illustrative example. Consider the network in Fig. 1. Activity $(2, 3)$ is distinguished. The network after the first, second and third call of Algorithm 3 is given in Figs. 5, 6 and

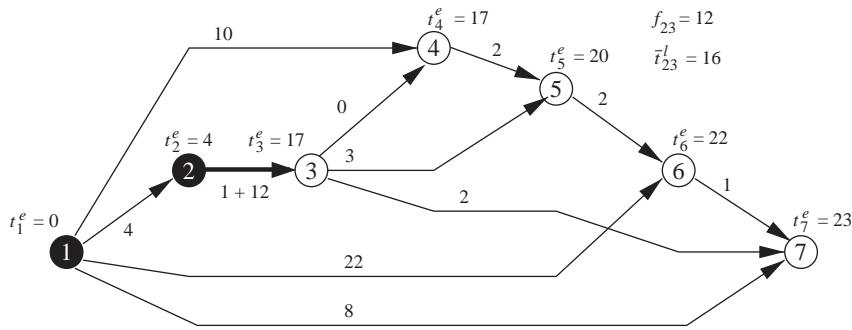


Fig. 7. The network, given in Fig. 1, after the third call of Algorithm 3—PHASE 2—in Algorithm 4.

7, respectively. Nodes labeled *true* after the calls of Algorithm 3 are black, activities whose free floats are computed in row 7 (see Algorithm 4) are marked with a dashed line, durations of activities that are at their upper bounds are in bold print. Activity (2, 3) is not necessarily critical with the durations equal to 1 and 6. For both, Algorithm 3 successfully determined the configurations in which there exists the critical path, $1 \rightarrow 6 \rightarrow 7$, that uses activities parallel to (2, 3) with the length greater than the lengths of all paths crossing (2, 3) (see Figs. 5, 6). (2, 3) becomes necessarily critical with the duration equal to 13, since the algorithm could not find such a configuration (see Fig. 7). The number $f_{23} = 12$, in Fig. 7, is the minimal one such that added to \underline{d}_{23} , makes (2, 3) necessarily critical. From this and Lemma 7, we obtain the maximal latest starting time $\bar{t}_{23}^l = 16$.

Lemma 8. *Algorithm 4 finds \bar{t}_{kl}^l .*

Proof. The proof goes in the similar manner to the one of Lemma 4. Namely, it suffices to show that f_{kl} , computed by Algorithm 4 (row 13), is the minimal nonnegative real number such that (k, l) with duration equal to $\underline{d}_{kl} + f_{kl}$ becomes necessarily critical. Then, making use of Lemma 7, we immediately obtain the maximal latest starting time of (k, l) .

Suppose, by contradiction, that there exists a nonnegative real number f'_{kl} such that (k, l) becomes necessarily critical with $\underline{d}_{kl} + f'_{kl}$ and satisfying the following inequality:

$$f'_{kl} < f_{kl}. \tag{14}$$

We denote the number of calls of Algorithm 3 in Algorithm 4 by q . This clearly forces $f_{kl} = \sum_{i=1}^{q-1} \Delta_i$ (see Algorithm 4, row 10). For the duration of (k, l) equals $\underline{d}_{kl} + \sum_{i=1}^{q-2} \Delta_i$, Algorithm 3 asserts that (k, l) is not necessarily critical. Furthermore, it returns a configuration of durations $\Omega', \Omega' \in \overline{\mathfrak{D}}_{kl}^s(\underline{d}_{kl} + \sum_{i=1}^{q-2} \Delta_i)$, node labels $label_i(\Omega')$, $i = 1, \dots, n$. For the duration of (k, l) equal to $\underline{d}_{kl} + f'_{kl}$, Algorithm 3 asserts the necessary criticality of (k, l) and returns a configuration of durations $\Omega'', \Omega'' \in \overline{\mathfrak{D}}_{kl}^s(\underline{d}_{kl} + f'_{kl})$, node labels $label_i(\Omega'')$, $i = 1, \dots, n$. Since (k, l) is necessary critical for $\underline{d}_{kl} + f'_{kl}$, but for $\underline{d}_{kl} + \sum_{i=1}^{q-2} \Delta_i$ (k, l) is not, we see $f'_{kl} > \sum_{i=1}^{q-2} \Delta_i$. Thus $f'_{kl} = \sum_{i=1}^{q-2} \Delta_i + \delta$ ($\delta > 0$). From the way of determining configurations Ω' and Ω'' by Algorithm 3 for $\underline{d}_{kl} + \sum_{i=1}^{q-2} \Delta_i$ and $\underline{d}_{kl} + \sum_{i=1}^{q-2} \Delta_i + \delta$, respectively, we

conclude that there exists at least one activity $(i', j') \in A$ such that

$$\begin{aligned} \text{label}_{i'}(\Omega') &= \text{false}, \text{label}_{j'}(\Omega') = \text{true}, \\ \text{label}_{i'}(\Omega'') &= \text{false}, \text{label}_{j'}(\Omega'') = \text{false} \end{aligned}$$

or

$$\text{label}_{j'}(\Omega') = \text{true}, \text{label}_{j'}(\Omega'') = \text{false} \text{ for } (i', j') = (k, l)$$

and satisfying

$$t_{j'}^e(\Omega') - t_{i'}^e(\Omega') - d_{i'j'}(\Omega') \leq \delta. \quad (15)$$

We now consider the stage after $(q - 2)$ th call of Algorithm 3. Algorithm 4 (rows 5 or 7) computes $\Delta_{q-1} = \min\{t_j^e(\Omega') - t_i^e(\Omega') - d_{ij}(\Omega') \mid (i, j) \neq (k, l), \text{label}_i(\Omega') = \text{false}, \text{label}_j(\Omega') = \text{true} \text{ or } (i, j) = (k, l), \text{label}_j(\Omega') = \text{true}\}$. From this and (15), it follows that $\Delta_{q-1} \leq \delta$. Consequently $\sum_{i=1}^{q-2} \Delta_i + \Delta_{q-1} \leq \sum_{i=1}^{q-2} \Delta_i + \delta$ and therefore $f_{kl} \leq f'_{kl}$, contrary to (14). \square

Lemma 9. *The running time of Algorithm 4 is $\mathcal{O}(mn)$.*

Proof. The proof proceeds in the same manner as the one of Lemma 5. In each call of Algorithm 3 at least one node is labeled *false*. Thus, Algorithm 3 ($\mathcal{O}(m)$) is invoked, in Algorithm 4, $\mathcal{O}(n)$ times (until $\text{label}(n) = \text{false}$). Hence, the complexity of Algorithm 4 is $\mathcal{O}(mn)$. \square

2.4. Determination of bounds on float of an activity

This section deals with the problem of determining the bounds on float of a given activity. That is, we consider the problems PLBF and PUBF, announced at the beginning of Section 2. We prove that one cannot approximate PLBF within a factor smaller than 1. Successively, we show that PLBF and PUBF are polynomially solvable in some special cases.

There are obvious connections between the notions of criticality and the bounds on float of an activity.

Proposition 6. *An activity $(k, l) \in A$ is possibly (resp. necessarily) critical in G if and only if $\underline{f}_{kl} = 0$ (resp. $\overline{f}_{kl} = 0$).*

From the above proposition, it may be concluded that PLBF is at least as hard, if not harder than the problem of the possible criticality of an activity and, similarly, PUBF is at least as hard as the problem of the necessary criticality of an activity. Accordingly, PLBF is strongly \mathcal{NP} -hard for general networks and remains \mathcal{NP} -hard even if the network G is restricted to be planar and regular of degree three (see [7]). Unfortunately, a question still unanswered is whether PUBF is polynomially solvable for general and planar networks. In [9] some heuristic methods for PLBF and PUBF have been proposed. At present, these problems are effectively solvable for networks having a special topology, namely the one of series-parallel graphs, for which the provided algorithms are polynomial [11].

We define the set of possible configurations of activity durations $\underline{\mathfrak{D}}_{kl}(x)$ (resp. $\overline{\mathfrak{D}}_{kl}(x)$), $x \geq 0$, to be the Cartesian product of corresponding time intervals D_{ij} , $(i, j) \in A$, given as follows:

$$D_{ij} = \begin{cases} [\underline{d}_{ij}, \overline{d}_{ij}] & \text{for } (i, j) \in \text{PREC}(k, l) \cup \text{SUCC}(k, l), \\ [x, x] & \text{for } (i, j) = (k, l), \\ [\underline{d}_{ij}, \underline{d}_{ij}] & \text{otherwise} \end{cases}$$

$$\left(D_{ij} = \begin{cases} [\underline{d}_{ij}, \overline{d}_{ij}] & \text{for } (i, j) \in \text{PREC}(k, l) \cup \text{SUCC}(k, l), \\ [x, x] & \text{for } (i, j) = (k, l), \\ [\overline{d}_{ij}, \overline{d}_{ij}] & \text{otherwise} \end{cases} \right).$$

The following proposition due to Dubois et al. [9].

Proposition 7. $f_{kl} = \min_{\Omega \in \underline{\mathfrak{D}}_{kl}(\overline{d}_{kl})} f_{kl}(\Omega)$ (resp. $\overline{f}_{kl} = \max_{\Omega \in \overline{\mathfrak{D}}_{kl}(\underline{d}_{kl})} f_{kl}(\Omega)$) Moreover, the minimum (resp. the maximum) $f_{kl}(\Omega)$ is attained on the vertices of the hyper-rectangle $\underline{\mathfrak{D}}_{kl}(\overline{d}_{kl})$ (resp. $\overline{\mathfrak{D}}_{kl}(\underline{d}_{kl})$).

It is worth pointing out that the reduction of the set of the possible configurations \mathfrak{D} to $\underline{\mathfrak{D}}_{kl}(\overline{d}_{kl})$ and to $\overline{\mathfrak{D}}_{kl}(\underline{d}_{kl})$ (Proposition 7) is not sufficient to keep PLBF from being \mathcal{NP} -complete and it does not allow to answer the question about the complexity of PUBF.

2.4.1. Inapproximability

The negative result for PLBL encourages us to look for approximation algorithms for this problem. Unfortunately, one cannot approximate PLBF within a factor smaller than 1.

Given a minimization problem P , let $opdp(I)$ denote the optimal solution value for some instance I of P and for a solution y of I , let $val_P(I, y)$ denote the associated value. Given a constant $\varepsilon \in (0, 1)$, a ε -approximation algorithm for P is an algorithm that applied to any instance I of P , runs in time bounded by polynomial in the size of I and produces a solution y whose value fulfills inequality $val_P(I, y) \leq \frac{1}{1-\varepsilon} opdp(I)$. If such an algorithm exists, we say that P is approximable within ε (see for instance [1]).

A negative approximation result is stated in the following theorem:

Theorem 1. *If $\mathcal{P} \neq \mathcal{NP}$ then it is not possible to approximate PLBF within $\varepsilon \in (0, 1)$, even when G is restricted to a planar network.*

Proof. See Appendix. \square

2.4.2. Some solvable special cases

We now study special cases of PLBF and PUBF for which polynomial algorithms for general networks exist.

Our basic assumption is that activities from set $\text{PREC}(k, l)$ have precise durations in network G , $\underline{d}_{ij} = \overline{d}_{ij}$ for all $(i, j) \in \text{PREC}(k, l)$, where $(k, l) \in A$ is a specified activity.

Lemma 10. Assume that activities $(i, j) \in PREC(k, l)$ have precise durations and f_{kl}^* is the minimal nonnegative real number such that (k, l) with a duration equal to $\bar{d}_{kl} + f_{kl}^*$ (resp. $\underline{d}_{kl} + f_{kl}^*$) becomes possibly (resp. necessary) critical. Then $\underline{f}_{kl} = f_{kl}^*$ (resp. $\bar{f}_{kl} = f_{kl}^*$).

Proof. From the assumption that activities $(i, j) \in PREC(k, l)$ have precise durations, we get $\mathfrak{D}_{kl}(x) = \mathfrak{D}_{kl}^s(x)$ (resp. $\bar{\mathfrak{D}}_{kl}(x) = \bar{\mathfrak{D}}_{kl}^s(x)$). Thus it suffices to show that $f_{kl}^* = \min_{\Omega \in \mathfrak{D}_{kl}^s(\bar{d}_{kl})} f_{kl}(\Omega)$ (resp. $f_{kl}^* = \max_{\Omega \in \bar{\mathfrak{D}}_{kl}^s(\underline{d}_{kl})} f_{kl}(\Omega)$). The proof of this equality runs in the same manner as the one of Lemma 3 (resp. Lemma 7). \square

Lemma 11. Assume that activities $(i, j) \in PREC(k, l)$ have precise durations. Then Algorithm 2 (resp. Algorithm 4) finds $\underline{f}_{kl}, \bar{f}_{kl} = \underline{t}_{kl}^l - \underline{t}_k^e$ (resp. $\bar{f}_{kl}, \bar{f}_{kl} = \bar{t}_{kl}^l - \bar{t}_k^e$).

Proof. Algorithm 2 (resp. Algorithm 4) finds \underline{t}_{kl}^l (resp. \bar{t}_{kl}^l) by Lemma 4 (resp. Lemma 8). That $\underline{t}_{kl}^l = \underline{t}_k^e + f_{kl}^*$ (resp. $\bar{t}_{kl}^l = \bar{t}_k^e + f_{kl}^*$), where f_{kl}^* is the minimal nonnegative real number such that (k, l) with a duration equal to $\bar{d}_{kl} + f_{kl}^*$ (resp. $\underline{d}_{kl} + f_{kl}^*$) becomes possibly (resp. necessary) critical over the set $\mathfrak{D}_{kl}^s(\bar{d}_{kl} + f_{kl}^*)$ (resp. $\bar{\mathfrak{D}}_{kl}^s(\underline{d}_{kl} + f_{kl}^*)$), follows from Lemma 3 (resp. Lemma 7). We thus get $f_{kl}^* = \underline{t}_{kl}^l - \underline{t}_k^e$ (resp. $f_{kl}^* = \bar{t}_{kl}^l - \bar{t}_k^e$). Activities $(i, j) \in PREC(k, l)$ have precise durations, and so $\mathfrak{D}_{kl}(x) = \mathfrak{D}_{kl}^s(x)$ (resp. $\bar{\mathfrak{D}}_{kl}(x) = \bar{\mathfrak{D}}_{kl}^s(x)$). Hence and Lemma 10 we conclude that $\underline{f}_{kl} = f_{kl}^*$ (resp. $\bar{f}_{kl} = f_{kl}^*$). \square

If activities $(i, j) \in SUCC(k, l)$ have precise durations, where $(k, l) \in A$ is a specified activity, then PLBF and PUBF are also polynomially solvable. It is sufficient to reverse the arcs in network G and apply Algorithms 2 and 4, respectively, with certain modifications in Algorithms 1 and 3 (see Remarks 1 and 2) together with Lemma 11.

3. The latest starting times and the 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 the activity durations, which are determined by means of fuzzy numbers \tilde{d}_{ij} , $(i, j) \in A$, which imprecisely determine durations of activities $(i, j) \in A$. Fuzzy number \tilde{d}_{ij} expresses uncertainty connected with the ill-known activity duration modeled by this number. It generates a possibility distribution for the sets of values containing the unknown activity duration. More formally, we say that the assertion of the form “ \mathfrak{d}_{ij} is \tilde{d}_{ij} ”, where \mathfrak{d}_{ij} is a variable and \tilde{d}_{ij} is a fuzzy number, generates the possibility distribution of \mathfrak{d}_{ij} with respect to the following formula (see [9,8]):

$$\Pi(\mathfrak{d}_{ij} = x) = \mu_{\tilde{d}_{ij}}(x), \quad x \in \mathbb{R}_+.$$

Let Ω be a configuration of activity durations in the network with activity times $d_{ij} \in \mathbb{R}_+$, $(i, j) \in A$. Thus, the (joint) possibility distribution over configurations, induced by the \tilde{d}_{ij} 's, is $\pi(\Omega) = \min_{(i,j) \in A} \mu_{\tilde{d}_{ij}}(d_{ij})$, $\Omega \in \mathbb{R}_+^m$. Hence, the possibility distribution describing possible values for the latest starting

times t_{kl}^l (resp. float \check{f}_{kl}) of an activity (k, l) is defined in following way (see [9]):

$$\begin{aligned} \mu_{\tilde{T}_{kl}^l}(x) &= \Pi(t_{kl}^l = x) = \sup_{\Omega: x=t_{kl}^l(\Omega)} \pi(\Omega), \quad x \in \mathbb{R}_+, \\ \mu_{\tilde{F}_{kl}}(x) &= \Pi(\check{f}_{kl} = x) = \sup_{\Omega: x=f_{kl}(\Omega)} \pi(\Omega), \quad x \in \mathbb{R}_+, \end{aligned}$$

where $t_{kl}^l(\Omega)$ (resp. $f_{kl}(\Omega)$) is the latest starting time (resp. the float) of (k, l) in configuration Ω .

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, $\tilde{T}_{kl}^l(\alpha)$ and $\tilde{F}_{kl}(\alpha)$, of each fuzzy latest starting time \tilde{T}_{kl}^l and float \tilde{F}_{kl} in a network with duration intervals $\tilde{d}_{ij}(\alpha) = [\underline{d}_{ij}(\alpha), \bar{d}_{ij}(\alpha)]$, $(i, j) \in A$. Then the fuzzy quantities, \tilde{T}_{kl}^l and \tilde{F}_{kl} , are reconstructed from their α -cuts. This approach makes sense since intervals $\tilde{T}_{kl}^l(\alpha) = [t_{kl}^l(\alpha), \bar{t}_{kl}^l(\alpha)]$ and $\tilde{F}_{kl}(\alpha) = [f_{kl}(\alpha), \bar{f}_{kl}(\alpha)]$ are nested. Such an approach has been proposed in [11,9] 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, the fuzzy latest starting time \tilde{T}_{kl}^l , i.e. its α -cuts, in general networks can be determined by means of the algorithm for computing the bounds on the latest starting times (Algorithms 2 and 4), described in Sections 2.2 and 2.3.

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 the bounds on floats is polynomially solvable.

4. Conclusion

In this paper, we have considered the problems of computing the bounds on the latest starting times and the floats of activities in a network with imprecise durations represented by means of interval or fuzzy numbers. We have proposed $\mathcal{O}(mn)$ time algorithms for computing the intervals of the possible values of the latest starting times of an activity in general networks with interval durations. So far, there have existed polynomial algorithms only for networks having a series–parallel topology. Furthermore, we have proved that the problem of computing the bounds on the floats of an activity in networks with interval durations cannot be approximated within a factor smaller than 1 and described some polynomially solvable cases.

We have also shown how to use the proposed algorithms for computing the fuzzy latest starting times of activities in a network with activity durations given in the form of fuzzy numbers.

Appendix

Proof of Theorem 1.

The proof is inspired by the \mathcal{NP} -completeness proof of the problem of the possible criticality of an activity, presented in [7] (Theorem 1). A polynomial reduction from a certain modified PARTITION problem, denoted MPARTITION, to the one of evaluating possible criticality was proposed.

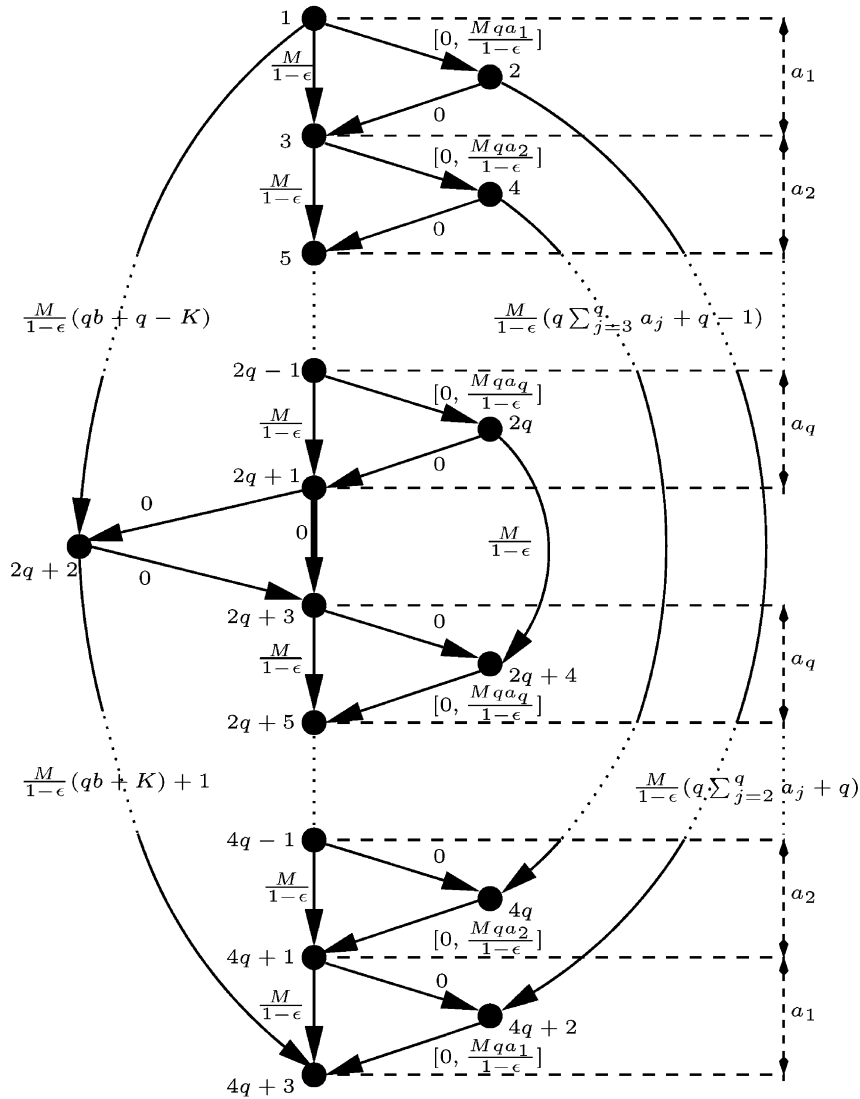


Fig. 8. The network G'' .

The MPARTITION is defined as follows:

INPUT: A finite set \mathcal{A} of positive integers, $\mathcal{A} = \{a_1, \dots, a_q\}$, having the overall sum of $2b$ (\mathcal{A} is allowed to have equal positive integers) and a positive integer $K < q$.

QUESTION: Is there a subset $\mathcal{A}' \subset \mathcal{A}$ that sums up exactly to b and $|\mathcal{A}'| = K$?

It is well known that MPARTITION is \mathcal{NP} -complete (see for instance [12] and comments on PARTITION given there).

We use the proof by contradiction. Suppose that for some $\epsilon < 1$, there is a ϵ -approximation algorithm for PLBF. The idea of the proof is to use the algorithm to construct a polynomial algorithm to solve MPARTITION. Since MPARTITION is \mathcal{NP} -complete, we get a contradiction if $\mathcal{P} \neq \mathcal{NP}$.

We now polynomially transform an instance of MPARTITION to an instance of PLBF for a planar network.

The transformation is divided into 2 steps. In the first one we associate to each instance of MPARTITION a directed, acyclic, planar network $G' = \langle A', V' \rangle$ with $4q + 3$ nodes (events) labeled $1, 2, \dots, 4q + 3$. The construction of G' is the same as in [7] (the proof of Theorem 1). In the second step we transform network G' , into $G'' = \langle A'', V'' \rangle$. That is, we multiply the interval bounds by $\frac{1}{1-\varepsilon}M$, where M is a sufficiently large number (for instance $M = q + 1$), and add 1 to the interval bounds of activity $(2q + 2, 4q + 3)$ (see Fig. 8). Activities $(2i, 2(2q - i + 2))$, $i = 1, \dots, q$, have time intervals $[\frac{M}{1-\varepsilon}(q \sum_{j=i+1}^q a_j + q - i + 1), \frac{M}{1-\varepsilon}(q \sum_{j=i+1}^q a_j + q - i + 1)]$, the one-point intervals have been written in Fig. 8 as precise times, $(2q + 1, 2q + 3)$ is a specified activity.

For graph G' , constructed in the first step, the following equivalence holds.

Claim 1. *There exists a subset $A' \subset A$ which sums up exactly to b and $|A'| = K$ if and only if the activity $(2q + 1, 2q + 3)$ is possibly critical in G' .*

By Proposition 6, we obtain another formulation (from the point of view of floats).

Claim 1'. *There exists a subset $A' \subset A$ which sums up exactly to b and $|A'| = K$ if and only if the lower bound on the float of activity $(2q + 1, 2q + 3)$ is zero.*

For graph G'' , constructed in the second step, the following claim may be proved:

Claim 2. *If there exists a subset $A' \subset A$ that sums up exactly to b and $|A'| = K$ then the lower bound on the float of activity $(2q + 1, 2q + 3)$ is equal to 1, $f_{2q+1, 2q+3} = 1$, in G'' . Otherwise it is at least $M/(1 - \varepsilon)$.*

The proof of Claim 2 proceeds in the same manner as Claim 1 (see the proof of Theorem 1 in [7]).

Consider what happens when we run the ε -approximation algorithm for PLBF on G'' . If there exists a subset $A' \subset A$ that sums up exactly to b and $|A'| = K$ then the algorithm returns the float of activity $(2q + 1, 2q + 3)$ whose value is at most $1/(1 - \varepsilon)$. Otherwise this value is at least $M/(1 - \varepsilon)$. In effect, we have given a polynomial time algorithm for MPARTITION. This contradicts the fact that MPARTITION is \mathcal{NP} -complete, unless $\mathcal{P} = \mathcal{NP}$. This completes proof. \square

References

- [1] G. Ausiello, P. Crescezi, M. Protasi, Approximate solution of NP optimization problems, Theoret. Comput. Sci. 150 (1995) 1–55.
- [2] J.J. Buckley, Fuzzy Pert, in: G.W. Evans, W. Karwowski, M.R. Wilhelm (Eds.), Applications of Fuzzy Set Methodologies in Industrial Engineering, Elsevier, Amsterdam-Oxford-New York-Tokyo, 1989, pp. 103–114.
- [3] S. Chanas, D. Dubois, P. Zieliński, On the sure criticality of tasks in activity networks with imprecise durations, IEEE Trans. Systems Man, Cybern. Part B: Cybern. 32 (2002) 393–407.
- [4] S. Chanas, J. Kamburowski, The use of fuzzy variables in PERT, Fuzzy Sets and Systems 5 (1981) 1–19.
- [5] S. Chanas, P. Zieliński, Critical path analysis in the network with fuzzy activity times, Fuzzy Sets and Systems 122 (2001) 195–204.
- [6] S. Chanas, P. Zieliński, The computational complexity of the criticality problems in a network with interval activity times, European J. of Oper. Res. 136 (3) (2002) 541–550.
- [7] S. Chanas, P. Zieliński, On the hardness of evaluating criticality of activities in a planar network with duration intervals, Oper. Res. Lett. 31 (2003) 53–59.

- [8] D. Dubois, H. Fargier, P. Fortemps, Fuzzy scheduling: Modelling flexible constraints vs. coping with incomplete knowledge, *European J. Oper. Res.* 147 (2003) 231–252.
- [9] D. Dubois, H. Fargier, V. Galvagnon, On latest starting times and floats in activity networks with ill-known durations, *European J. Oper. Res.* 147 (2003) 266–280.
- [10] D. Dubois, H. Prade, *Fuzzy Sets and Systems: Theory and Applications*, Academic Press, New York, 1980.
- [11] H. Fargier, V. Galvagnon, D. Dubois, Fuzzy PERT in series–parallel graphs, 9th IEEE Internat. Conf. Fuzzy Systems, San Antonio, TX, 2000, pp. 717–722.
- [12] M.R. Garey, D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman and Co., San Francisco, 1979.
- [13] I. Gazdik, Fuzzy network planning, *IEEE Trans. Reliability* 32 (1983) 304–313.
- [14] M. Hapke, A. Jaskiewicz, R. Słowiński, Fuzzy project scheduling system for software development, *Fuzzy Sets and Systems* 67 (1994) 101–117.
- [15] A. Kaufmann, M.M. Gupta, *Fuzzy Mathematical Models in Engineering and Management Science*, North-Holland, Amsterdam, 1988.
- [16] J.E. Kelley, M.R. Walker, Critical path planning and Scheduling, in: *Proc. Eastern Joint Comput. Conf.* 16 (1959) 160–172.
- [17] C.S. McCahon, Using PERT as an Approximation of Fuzzy Project-Network Analysis, *IEEE Trans. Eng. Manage.* 40 (1993) 146–153.
- [18] C.S. McCahon, E.S. Lee, Project network analysis with fuzzy activity times, *Comput. Math. Appl.* 15 (1988) 829–838.
- [19] S.H. Nasution, Fuzzy Critical Path Method, *IEEE Trans. Systems Man, Cybernetics* 24 (1994) 48–57.
- [20] H. Prade, Using fuzzy sets theory in a scheduling problem: a case study, *Fuzzy Sets and Systems* 2 (1979) 153–165.
- [21] H. Rommelfanger, Network analysis and information flow in fuzzy environment, *Fuzzy Sets and Systems* 67 (1994) 119–128.