# On Computing the Minimal Latest Starting Times of Activities in Interval-Valued Networks with Generalized Precedence Relations

Siamak Haji Yakhchali

*Abstract*— **These instructions give you guidelines for preparing papers for WCE. Use this document as a template if you are using Microsoft Word 6.0 or later. Otherwise, use this document as an instruction set. The electronic file of your paper will be formatted further at IAENG. Define all symbols used in the abstract. Do not cite references in the abstract. Do not delete the blank line immediately above the abstract; it sets the footnote at the bottom of this column.**

*Index Terms*— **Uncertainty, fuzzy intervals, scheduling, project management**

## I. Introduction

The critical path method (CPM) [17] has become one of the tools that is the most useful in practice and control of complex projects. The critical path analysis is based on the computation of latest starting times of activities from the knowledge of the earliest ending time of the project. Then paths containing critical activities, that are activities with zero floats, are identified. What is essential in the CPM method is that the activity durations are deterministic and known. However, the operation time for each activity is usually difficult to define and estimate precisely in a real situation.

Program Evaluation and Review Technique (PERT) [21] which tries to deal with uncertainty assumes that the durations of the activities are random variable with the beta distribution, aim at providing the distribution function of the total duration . So far, in the literature, hundreds of papers have used this stochastic approach and search on this area is still carried out (e.g. [13], [14]). Anyway, these stochastic methods rely on statistical data which are out of reach in many cases [20], and on dubious independence assumptions [9].

Since the pioneering work of Zadeh [32], other researchers in this field have recommended the use of fuzzy numbers for modeling activity durations rather than stochastic variables. Fuzzy critical path methods and fuzzy PERT have been proposed since the late 1970s (e.g. [16],[23]). The possible values of the earliest starting times can be computed by means of a forward recursion procedure comparable to the one used in traditional CPM problems. Unfortunately, the backward recursion issued from CPM is indeed not sound if durations are described by means of fuzzy intervals, in fact, the backward recursion takes the imprecision of some duration twice into account [11]. Zielinski [33] completely determined the possible values of the latest starting times of a given activity. Fortin et al. [15] proposed an algorithm for determining the minimal latest starting times of activities and computed the maximal float of an activity. Yakhchali and Ghodsypour [27] suggested a hybrid genetic algorithm for the minimal float of an activity. Dubois et al. [10] have proposed an efficient algorithm based on path enumeration to compute optimal intervals for latest starting times and floats of activities.

The criticality concept in networks with interval (fuzzy) activities durations is a more realistic approach than the traditional ones. Instead of being critical or not, the activities or paths that are for sure critical despite uncertainty is called necessarily critical, those that are for sure not critical is called necessarily noncritical and those whose criticality is unknown, called possibly critical [3]. The idea of partitioning is used by Yakhchali and Ghodsypour [29] to develop an algorithm for determining these three type of critical activity. The possibilistic criticality analysis is carried out by Chanas and Zielinski [5] for interval-valued durations, and Chanas and Zielinski [4] for fuzzy durations. The problems of the necessarily and possibly critical paths in the networks with imprecise activity and time lag durations have been discussed by Yakhchali et al. [25], [26].

The traditional precedence relation, suggested by the CPM model, is the finish-start precedence relation with zero time lag. In practice it is often necessary to specify other than this precedence relation. In accordance with [12], we will refer to the resulting types of precedence relations as generalized precedence relations (GPRs). We distinguish between four types of GPRs: start-start (SS), start-finish (SF), finish-start (FS) and finish-finish (FF). GPRs can specify a minimal or a maximal time lag between a pair of activities. A minimal time lag specifies that an activity can only start (finish) when the predecessor activity has already started (finished) for a certain time period. A maximal time lag specifies that an activity should be started (finished) at the latest, within a certain number of time periods beyond the start (finish) of another activity [6]. GPRs can be used to model a wide variety of specific problem characteristics, including: activity ready times and deadlines, activities that have to start or terminate simultaneously, non-delay execution of activities, several types (total or strong/week partial) of mandatory activities overlaps, fixed activity start times, time-varying resource requirement and availabilities, time-windows for resources, inventory restrictions, set-up times, overlapping production activities, assembly line zoning constraints, etc [8] and [22].

Siamak Haji Yakhchali, Department of Industrial Engineering, Faculty of Engineering, University of Tehran, Tehran, Iran; e-mail: yakhchali@ut.ac.ir.

ISBN: 978-988-18210-6-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Yakhchali and Ghodsypour proposed a series studies on the topic of the project scheduling problems in the networks with GPRs and imprecise durations. They have completely solved the problem of determining the possible value of the latest starting time of an activity in acyclic networks [28] and cyclic networks [30] with GPRs and imprecise durations. They suggested an algorithm for computing both the possible values of the latest starting times and the floats of all activities in networks with generalized precedence relations and imprecise durations [31]. In this paper, We complete these researches by proposing a novel polynomial algorithm for computing the minimal latest starting times of all activities in general networks with GPRs and interval durations.

## II. The Project Scheduling Problems in Interval-Valued Networks with GPRs

A network *G=<V, E>* with node set *V*, arc set *E,* being a project activity-on-node (AON) model, is given. A set *V={1,2, …, n}* of *activities* has to be executed where activity durations $i \in V$ are chosen from intervals $D_i = [\underline{d}_i, \overline{d}_i]$, $\underline{d}_i \ge 0$. The non-preemptable activities are numbered from *1* to *n*, where the dummy activities *1* and *n* represent the beginning and the termination of the project, respectively. Dummy activities are only needed to satisfy the requirement that the network possesses only one initial and one terminal node.

The arc set or generalized precedence relations, *E*, consist of *minimal or maximal time lag*. If between two activities *i* and *j* a minimal or maximal time lag is prescribed, we introduce an arc (*i, j*) from node *i* to node *j* weighted by an interval number which have the forms:

$$s_i + SS_{ij}^{min} \le s_j \le s_i + SS_{ij}^{max}$$

$$s_i + SF_{ij}^{min} \le f_j \le s_i + SF_{ij}^{max}$$

$$f_i + FS_{ij}^{min} \le s_j \le f_i + FS_{ij}^{max}$$

$$f_i + FF_{ij}^{min} \le f_j \le f_i + FF_{ij}^{max}$$

where the start of an activity is given by $s_i$ and its finishing time denoted by $f_i$. $SS_{ij}^{min}$ represents a minimal time lag between the start time of activity *i* and the start time of activity *j* and the value of $SS_{ij}^{min}$ are chosen from intervals $SS_{ij}^{min} = [\underline{ss}_{ij}^{min}, \overline{ss}_{ij}^{min}]$ (similar definition apply for $SS_{ij}^{max}$, $SF_{ij}^{min}$, …).

The various time lags can be represented in a *standardized form* by transforming them to, for instance, minimal SS precedence relation, using the transformation rules [30] as following:

$$s_i + SS_{ij}^{min} \le s_j \Rightarrow s_i + L_{ij} \le s_j \quad \text{with } L_{ij} = [\underline{ss}_{ij}^{min}, \overline{ss}_{ij}^{min}]$$

$$s_i + SS_{ij}^{max} \le s_j \Rightarrow s_j + L_{ji} \le s_i \quad \text{with } L_{ji} = [-\overline{ss}_{ij}^{max}, -\underline{ss}_{ij}^{max}]$$

$$s_i + SF_{ij}^{min} \le f_j \Rightarrow s_i + L_{ij} \le s_j \quad \text{with} L_{ij} = [\underline{sf}_{ij}^{min} - \overline{d}_j, \overline{sf}_{ij}^{min} - \underline{d}_j]$$

$$s_i + SF_{ij}^{max} \le f_j \Rightarrow s_j + L_{ji} \le s_i \quad \text{with} L_{ji} = [\underline{d}_j - \overline{sf}_{ij}^{max}, \overline{d}_j - \underline{sf}_{ij}^{max}]$$

$$f_i + FS_{ij}^{min} \le s_j \Rightarrow s_i + L_{ij} \le s_j \quad \text{with } L_{ij} = [\underline{d}_i + \underline{fs}_{ij}^{min}, \overline{d}_i + \overline{fs}_{ij}^{min}]$$

$$f_i + FS_{ij}^{max} \le s_j \Rightarrow s_j + L_{ji} \le s_i \quad \text{with} L_{ji} = [-\overline{d}_i - \overline{fs}_{ij}^{max}, -\underline{d}_i - \underline{fs}_{ij}^{max}]$$

$$f_i + FF_{ij}^{min} \le f_j \Rightarrow s_i + L_{ij} \le s_j \quad \text{with} L_{ij} = [\underline{d}_i - \overline{d}_j + \underline{ff}_{ij}^{min}, \overline{d}_i - \underline{d}_j + \overline{ff}_{ij}^{min}]$$

$$f_i + FF_{ij}^{max} \le f_j \Rightarrow s_j + L_{ji} \le s_i \quad \text{with} L_{ji} = [\underline{d}_j - \overline{d}_i - \overline{ff}_{ij}^{max}, \overline{d}_j - \underline{d}_i - \underline{ff}_{ij}^{max}]$$

In this way, all GPRs are consolidated in the expression $s_i + L_{ij} \le s_j$. Project networks with GPRs can be represented as cyclic networks. A positive path length from a node to itself indicates the existence of a cycle of positive length, and consequently, the non-existence of a time feasible schedule.

To clarify the transformation rules, we provide an example. Fig. 2 shows the standardized form of a network in Fig. 1 that was proposed by De Reyck [7]. In Fig. 1, the interval number adjacent to each node represents the corresponding activity duration and the labels associated with the arcs indicate the interval GPRs

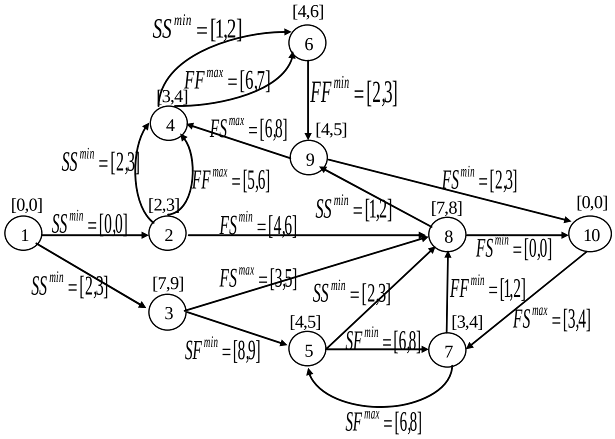


Fig. 1. An activity network with GPRs (De Reyck [7]

In Fig. 2, the interval number along each arc denotes the interval time lag. If there is more than one time lag between two activities *i* and *j*, only the maximal time lag is retained. For example, there are two time lags between activities 5 and 7, so the maximal time lag, $l_{5,7} = [2,5]$, is considered (the other is shown in a dash line).

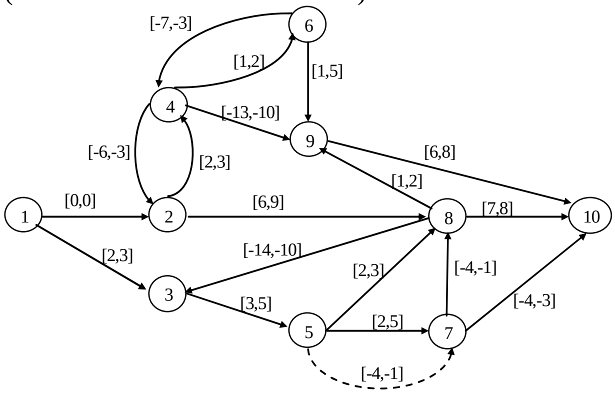


Fig. 2. A The standardized form of the network in Fig. 1

As far as the standardized form is concerned, the minimum time lag, denoted by $L_{ij} = [\underline{l}_{ij}, \overline{l}_{ij}]$, *i, j* $\in$ *V*, implies that *j* can start $l_{ij}$ which is chosen from intervals $[\underline{l}_{ij}, \overline{l}_{ij}]$ units of time after the start of *i* at the earliest. In the remainder of this paper, the network *G* is assumed in its standardized form.

The notation of *configuration* denoted by $\Omega$ has been defined by Buckley [2] to relate the interval case to the

ISBN: 978-988-18210-6-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

deterministic case of classical PERT/CPM problems. We redefine a configuration as a tuple of time lag durations, $(l_{12},...,l_{ij})$, such that $\forall (i,j) \in E$, $l_{ij} \in L_{ij}$. For a configuration $\Omega$, $l_{ij}(\Omega)$ will denote the duration of time lag (*i*, *j*). The pessimistic configuration, denoted by $\overline{\Omega}$, is a configuration $\overline{\Omega} \in \omega$ that $l_{ij}(\overline{\Omega}) = \bar{l}_{ij}$ for all $(i,j) \in E$ and similarly $\underline{\Omega}$, called the optimistic configuration, is a configuration $\underline{\Omega} \in \omega$ that $l_{ij}(\underline{\Omega}) = \underline{l}_{ij}$ for all $(i,j) \in E$.

Let us denote the set of all paths in *G* from node "*1*"to "*n*" by *P*. $P(i,j)$ will denote the set of all paths in the subnetwork $G(i,j)$. A subnetwork of *G*, $G(i,j)$, composes of nodes succeeding *i* and preceding *j*, $i,j \in V$.

The configuration related to a path, *p*, will defined as the formula (1). We call this configuration a path-induced configuration, denoted by $\Omega_p$.

$$l_{ij}(\Omega) = \begin{cases} \bar{l}_{ij} & \text{for}\,(i,j) \in p \\ \underline{l}_{ij} & \text{for}\,(i,j) \notin p \end{cases} \qquad (1)$$

$Pred(j)$ is the set of immediate predecessors of an activity *j*, *j*∈ *V*: $Pred(j) = \{i \mid (i,j) \in E\}$ and $Succ(i)$ is the set of immediate successors of an activity *i*, *i*∈ *V*: $Succ(i) = \{j \mid (i,j) \in E\}$.

$s_i^e(\Omega)$ and $s_i^l(\Omega)$ will denote the earliest starting time and the latest starting time of activity *i* in the configuration $\Omega$, respectively. The value of $s_i^e(\Omega)$ can be calculated by finding the longest path from node '*1*' to node '*i*'. So standard graph algorithms for computing the longest paths in networks, for example Floyd-Warshall algorithm (time complexity $O(|V|^3)$), can be used (see [19]). The earliest starting time can be computed more efficiently by using the Modified Label Correcting Algorithm [1] (MLCA) whose complexity is $O(|V||E|)$.

*Proposition 1* ([28]): The optimistic configuration, $\underline{\Omega}$, minimizes the earliest starting time of all the activities, $k \in V$, and the pessimistic configuration, $\overline{\Omega}$, maximizes their earliest starting times.

Thus, it is enough to use the Modified Label Correcting algorithm to compute the bounds on the earliest starting times of all activities. Unfortunately the optimistic and pessimistic configurations fail to compute the minimal and maximal of the latest starting times of activities. The trouble with computing the possible values of the latest starting times is explained by several authors (e.g. Dubois et al. [11],[10] and Zielinski [33])

In the following section a polynomial algorithm is proposed for computing the minimal latest starting times of all activities.

## III. The Polynomial Algorithm for Computing the Minimal Latest Starting Times

In the case of interval durations, the minimal latest starting times for a given activity. denoted by $\underline{s}_k^l$, is equal $\underline{s}_k^l = min_{\Omega \in \omega} s_k^l(\Omega)$ where $\omega$ is the set of possible configurations of time lag durations

*Proposition 2* ([31]): There exists a path *p*, $p \in P(k,n)$, that the path-induced configuration, $\Omega_p$, minimizes the latest starting time of *k*, $\underline{s}_k^l = s_k^l(\Omega_p)$.

The key to constructing the algorithm for computing the minimal latest starting times of activities is Proposition 3. The idea of Algorithm 1 is based on this proposition. It consists in finding path *p*, $p \in P(l,n)$ that added to $(k,l)$ build a configuration $\Omega_p$ which $\underline{s}_k^l = s_k^l(\Omega_p)$.

*Proposition 3*: $\underline{s}_k^l = min_{l \in Succ(k)} \{ s_k^l(\Omega_{\{(k,l)\} \cup p}) \}$ where $p \in P(l,n)$ and $\underline{s}_l^l = s_l^l(\Omega_p)$.

*Proof:* Based on Proposition 2, there exists a path *p*, $p \in P(k,n)$, that the path-induced configuration, $\Omega_p$, minimize the latest starting time of *k*, $\underline{s}_k^l = s_k^l(\Omega_p)$. It is worth pointing out that *p* is one of the longest paths from *k* to *n* in $\Omega_p$. According Proposition 2, there exists a $p'$, $p' \in P(l,n)$ and $l \in Succ(k)$ which path-induced configuration by $p'$, $\Omega_{p'}$, minimizes the latest starting time of *l*, $\underline{s}_l^l = s_l^l(\Omega_p)$ and $p'$ is one of the longest paths from *l* to *n* in $\Omega_{p'}$. Thus, the path *p* consists of $(k,l)$, $l \in Succ(k)$, and $p'$, $p' \in P(l,n)$, in fact $p = \{(l,n)\} \cup p'$. □

**Algorithm 1**: Computing the minimal latest starting times of activities

Input: A network $G = \langle V,E \rangle$, time lag intervals $L_{ij} = [\underline{l}_{ij}, \bar{l}_{ij}]$ $\forall (i,j) \in E$
Output: The minimal latest starting times of all the activities in *G*

**for** $i \in V$ **do**
... $\underline{s}_i^l \leftarrow +\infty$
**end for**
Call MCLA and compute the $s_n^l(\underline{\Omega})$
$\underline{s}_n^l = s_n^l(\underline{\Omega})$
$p_n \leftarrow \{\}$
$List \leftarrow \{n\}$
**while** $List \neq \phi$ **do**
...**for** $j \in List$ **do**
......**for** $i \in Pred(j)$ **do**
......... $p' \leftarrow \{(i,j)\} \cup p_j$
.........Call MCLA and compute the $s_i^l(\Omega_p)$
.........**if** $s_i^l(\Omega_p) < \underline{s}_i^l$ **then**
............ $\underline{s}_i^l \leftarrow s_i^l(\Omega_p)$
............ $p_i \leftarrow p'$
............**if** $i \notin List$ **then** $List \leftarrow List \cup \{i\}$ **end if**
.........**end if**
......**end for**
...... $List \leftarrow List - \{j\}$
...**end for**
**end while**

Algorithm 1 computes the minimal latest starting times of activities by using MCLA on each path-induced configuration based on Proposition 3. At each iterations, the algorithm assigns tentative minimal latest starting times to activities. These values are estimates of the longest path length and are considered as temporary until the final iteration, where $List = \phi$.

ISBN: 978-988-18210-6-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

## IV. Complexity

Proposition 4 shows that the proposed algorithm is polynomial.

*Proposition 4*: The running time of Algorithm 1 is $O(|V|^2.|E|^2)$

*Proof:* The running times of Modified Label Correcting Algorithm (MLCA), the best known algorithm for temporal analysis in cyclic project networks, is $O(|V||E|)$. The line 8 to 21, *while loop*, executed $|V|$ times and the line 10 to 18, *for-for end*, executed $|E|$ times, so the overall running time is $O(|V|^2.|E|^2)$.☐

Table 1 represent the summary of the complexity of the different problems in networks with generalized precedence relations and interval activity and time lag durations.

TABLE 1
THE RUNNING TIMES OF THE BEST KNOWN ALGORITHMS FOR INTERVAL PROJECT SCHEDULING IN NETWORKS WITH GPRS

| Problems | Minimal/Maximal | Running time | Ref. |
|---|---|---|---|
| Earliest starting times (all activities) | Minimal<br>Maximal | $O(\|V\|.\|E\|)$ | [28] |
| Latest starting time (an activity) | Minimal<br>Maximal | $O(\|V\|^2.\|E\|^2)$<br>$O(\|V\|^3.\|E\|^3)$ | [30] |
| Total floats (all activities) | Minimal<br>Maximal | $O(\|V\|^2.\|E\|.\|P\|)$ | [31] |
| Latest starting time (all activities) | Minimal | $O(\|V\|^2.\|E\|^2)$ | Algorithm 1 |

It is worth noticing that some algorithm, e.g. Algorithm 1, need only one execution for computing a given characteristic for all activities of a network. On the other hand, others need to be executed for each activity.

## V. Computational Experience

Let us clarify the utility of the proposed algorithm on some realistic project networks. For this reason, the minimal latest starting times of activities of project networks that have been generated by ProGenMax [24]. Schwindt [24] developed ProGenMan based on the problem generator ProGen [18]. Kolisch and Sprecher [18] are supposed to be representative of real project scheduling problems. On those problems, activities durations are precisely defined, thus we have added a relative uncertainty range of 20% to obtain intervals.

Table 2 presents the performance of the proposed algorithm on libraries of project networks, with respectively, 12, 22, 32, 52, 102 and 202 activities (on 360, 360, 270, 1170, 90 and 90 instances of project networks, respectively). Hence, we have determined the interval of the latest starting times of 163080 activities in 2340 project networks. The proposed algorithm has been programmed in MATLAB (R2006b) and run on a personal computer with 1.60 GHz processor (Intel Centrino 1.7) and 512 MB of RAM. The overall execution times, expressed in seconds, are measured. The tested networks can be downloaded from the web site

[http://www.wior.uni-karlsruhe.de/LS_Neumann/Forschung/ProGenMax/rcpspmax.html](http://www.wior.uni-karlsruhe.de/LS_Neumann/Forschung/ProGenMax/rcpspmax.html).

TABLE 2
THE EXECUTION TIMES OF ALGORITHM 1

| | | | Execution time | | |
|---|---|---|---|---|---|
| Testsets name | Nb of act. | Nb of net. | Min. | Ave | Max. |
| SM-J10 | 12 | 270 | 0.0732 | 0.1892 | 1.5185 |
| SM-J20 | 22 | 270 | 0.1024 | 0.3823 | 1.4913 |
| SM-J30 | 32 | 270 | 0.2696 | 1.2074 | 4.1159 |
| UBO10 | 12 | 90 | 0.1001 | 0.175 | 0.345 |
| UBO20 | 22 | 90 | 0.373 | 1.1739 | 2.64 |
| UBO50 | 52 | 90 | 1.8006 | 8.9072 | 28.21 |
| UBO100 | 102 | 90 | 30.91 | 190.42 | 930.63 |
| UBO200 | 202 | 90 | 598.6 | 1942.7 | 8566.3 |
| Testsets C | 102 | 540 | 4.1714 | 39.1126 | 289.0019 |
| Testsets D | 102 | 540 | 6.2319 | 29.2679 | 150.1842 |

In Table 2, the test sets name, number of activities in each test sets, number of tested networks, minimal, average and maximal execution times of the algorithm in second are shown in columns, respectively. Table 3 compares the performance of proposed algorithm with the recent path-algorithm proposed by Yakhchali and Ghodsypour [31] in the same test problems.

TABLE 3
THE COMPARISON BETWEEN EXECUTION TIMES OF ALGORITHM 1 AND ALGORITHM IN [31]

| | Algorithm 1 execution | | | Path algorithm excution | | |
|---|---|---|---|---|---|---|
| Testsets name | Min. | Ave. | Max. | Min. | Ave. | Max. |
| SM-J10 | 0.0732 | 0.1892 | 1.5185 | 0.0322 | 0.2238 | 4.3486 |
| SM-J20 | 0.1024 | 0.3823 | 1.4913 | 0.1271 | 37.5291 | 1348.26 |
| SM-J30 | 0.2696 | 1.2074 | 4.1159 | 0.3378 | 543.534 | 32526.7 |
| UBO10 | 0.1001 | 0.175 | 0.345 | 0.0364 | 0.1714 | 0.9273 |
| UBO20 | 0.373 | 1.1739 | 2.64 | 0.1599 | 23.2574 | 374.063 |
| UBO50 | 1.8006 | 8.9072 | 28.21 | 9.7412 | 2586.38 | 23476.4 |

There is no doubt that the proposed algorithm should be better.

## VI. Conclusion

This paper aims at computing the minimal latest starting times of all activities in networks with generalized precedence relations and interval activity and time lag durations. After transforming the interval-valued network into standardized form, the proposed algorithm can be used for determining the minimal latest starting times of all activities. Extensive computational results are reported using a problem set consisting of 2340 instances with up to 200 activities. The experimental results have been compared with the former algorithms. Hence, the proposed algorithm can be used in future project planner software that will cope with uncertainty.

As mentioned by several authors, the main difficulty of determining fuzzy project characteristics is the interval valued case and does not lie on the introduction of fuzzy sets. Thus, the proposed algorithm can be extended to networks with fuzzy durations.

## References

[1] D. R. K. Ahuja, T. L. Magnanti, J. B. Orlin, Networks Flows, In G. L. Nemhauser, A. H. G. Rinnooy Kan, M. J. Todd (Eds), Handbooks in Operations Research and Management Science, Elsevier, Amsterdam, (1989) 258-263.

[2] J. J. Buckley, Fuzzy PERT, In G. W. Evans, W. Karwowski, M. Wilhelm (Eds), Application of Fuzzy Set Methodologies in Industrial Engineering, Elsevier, Amsterdam, (1989) 103-125.

ISBN: 978-988-18210-6-5
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

[3] S. Chanas, D. Dubois, P. Zielinski, On the sure criticality of tasks in activity networks with imprecise durations, IEEE Transaction on Systems, Man, and Cybernetics-Part B 32 (2002) 393-407.
[4] S. Chanas, P. Zielinski, Critical Path analysis in the network with fuzzy activity times, Fuzzy Set and Systems 122 (2001) 195-204.
[5] S. Chanas, P. Zielinski, The computational complexity of the criticality of the problems in a network with interval activity times, European Journal of Operational Research 136 (3) (2002) 541-550.
[6] E. L. Demeulemeester, W. S. Herroelen, Project Scheduling: A Research Handbook. Kluwer Academic Publishers (2002).
[7] B. De Reyck, Scheduling projects with generalized precedence relations: Exact and heuristic approaches, Ph.D. Thesis, Department of Applied Economics, Leuven University, Belgium (1998).
[8] B. De Reyck, W. Herroelen, A branch-and-bound procedure for the resource-constrained project scheduling problem with generalized precedence relations, European Journal of Operational Research 111 (1998) 152-174.
[9] B. Dodin, Bounding the project completion time distribution in PERT networks, Operations Research 33 (1985) 862-881.
[10] D. Dubois, H. Fargier, J. Fortin, Computational methods for determining the latest starting times and floats of tasks in interval-valued networks, Journal of Intelligent Manufacturing 16 (2005) 407-421.
[11] 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.
[12] S. E Elmaghraby, J. Kamburowski, The Analysis of Activity Network under Generalized Precedence Relations (GPRs), Management Science 38 (1992) 1245-1263.
[13] S.M.T. Fatemi Ghomi, E. Teimouri, Path critical index and task critical index in PERT networks, European Journal of Operational Research 141 (2002) 147–152.
[14] S.M.T. Fatemi Ghomi, M. Rabbani, A new structural mechanism for reducibility of stochastic PERT networks, European Journal of Operational Research 145 (2003) 394–402.
[15] J. Fortin, P. Zielinski, D. Dubois, H. Fargier, Interval analysis in scheduling, in: Principles and Practice of Constraint Programming - CP 2005 11th International Conference, CP 2005, P. van Beek (Ed.), LNCS 3709, Springer Verlag, Berlin (2005) 226–240.
[16] I. Gazdik, Fuzzy network planning, IEEE Transaction on Reliability R-32 3 (1983) 304-313.
[17] J. E. Kelley, M. R. Walker, Critical path planning and scheduling, Eastern Joint Computer Conference 16 (1959) 160-172.
[18] R. Kolisch, A. Sprecher, Psplib – a project scheduling library, European Journal of Operational Research 96 (1996) 205–216.
[19] E. L. Lawler, Combinatorial Optimization: Networks and Matroids, Holt, Rinehart, and Winston, New York (1976).
[20] F. A. Lootsma, Stochastic and fuzzy PERT, European Journal of Operational Research 43 (1989) 174-183.
[21] D. G. Malcolm, J. H. Roseboom, C. E. Clark, Application of a technique for research and development program evaluation, Operations Research 7 (1959) 646-669.
[22] K. Neumann, C. Schwindt, Activity-on-node networks with minimal and maximal time lags and their application to make-to-order-production, OR Spectrum 19 (1997) 205-217.
[23] H. Prade, Using fuzzy sets theory in a scheduling problem: a case study, Fuzzy Sets and Systems 2 (1979) 153-165.
[24] C. Schwindt, Generation of Resource-Constrained Project Scheduling Problems Subject to Temporal Constraints, Technical Report WIOR-543, Karlsruhe university (1998).
[25] S. H. Yakhchali, M. H. Fazel Zarandi, I. B. Turksen, S. H. Ghodsypour, Possible criticality of paths in networks with imprecise durations and time lags, IEEE conference, NAFIPS (2007) 277-282.
[26] S. H. Yakhchali, M. H. Fazel Zarandi, I. B. Turksen, S. H. Ghodsypour, Necessary criticality of paths in networks with imprecise durations and time lags, IEEE conference, NAFIPS (2007) 271-276.
[27] S. H. Yakhchali, S. H. Ghodsypour, Hybrid genetic algorithms for computing the float of activities in networks with imprecise durations, IEEE international conference on fuzzy systems (FUZZ-IEEE), Hong Kong (2008) 1789-1794.
[28] S. H. Yakhchali, S. H. Ghodsypour, Computing the latest starting times of activities in interval-valued networks with minimal time lags, European Journal of Operational Research, submitted.
[29] S. H. Yakhchali, S. H. Ghodsypour, On the latest starting times and criticality of activities in a network with imprecise durations, Applied Mathematical Modelling, submitted.
[30] S. H. Yakhchali, S. H. Ghodsypour, On latest starting times of activities in networks with imprecise durations and generalized precedence relations, Computers and Industrial Engineering, submitted.
[31] S. H. Yakhchali, S. H. Ghodsypour, A Path Enumeration Approach for Temporal Analysis in Networks with imprecise durations and generalized precedence relations, Applied Soft Computing, submitted.
[32] L. A. Zadeh, Fuzzy sets as a basis for a theory of possibility, Fuzzy Sets and Systems 1 (1978) 3-28.
[33] P. Zielinski, On computing the latest starting times and floats of activities in a network with imprecise durations, Fuzzy Sets and Systems 159 (2005) 53-76.