A Minimal Dominant Set of Critical Paths for the Uncertain Project - Network

Yuri N. Sotskov[1], Svetlana, Frank Werner[2*]
[1]United Institute of Informatics Problems,
National Academy of Sciences of Belarus,
Surganova Str. 6, Minsk, 220012, Belarus
sotskov@newman.bas-net.by

[2]Fakultät für Mathematik, Otto-von-Guericke-Universität Magdeburg,
Postfach 4120, 39016 Magdeburg, Germany
frank.werner@ovgu.de

Abstract

In project management, it is usually difficult to obtain the exact values of the activity durations and the assumption is more realistic that the activity duration may remain uncertain until the activity completion. We assume that lower and upper bounds on a factual activity duration are given at the stage of project planning, the probability distribution of a random duration being unknown before the activity completion. Therefore, one cannot find a priory a critical path in the given project-network G. We propose a two-step approach, where the initial project-network G is minimized in the first step and the resulting minimized project-network determines a minimal dominant set of the critical paths in the second step. A fuzzy logic procedure (or another heuristic technique) may be used to choose a single potentially critical path from the minimal dominant set.

Keywords: project management, uncertain activity, dominant paths

1. Introduction

This paper addresses project management with uncertain (interval) activity durations. We use the terminology from [8] for graph theory and that from [7] for scheduling theory.

In project management [1, 2, 5, 6], it is usually difficult to obtain the exact values of the activity durations and the assumption is more realistic that the activity duration may remain uncertain until the completion of the activity [2]. We assume that lower and upper bounds on a factual activity duration are given at the stage of project planning, the probability distribution of a random duration being unknown before the completion of the activity [2, 3, 4]. Therefore, one cannot find a priory a critical path in the given project-network $G = (V; A)$. We propose a two-step approach, where the initial project-network G is minimized in the first step and the resulting minimized project-network determines a minimal dominant set of critical paths in the second step. A fuzzy logic procedure (or

* corresponding author.

another heuristic technique) may be used to choose a unique potentially critical path from the minimal dominant set.

The rest of the paper is organized as follows. The problem setting and some notations are given in Section 2. Criteria for the dominance of a path by another path are proven in Section 3. Properties of so-called a-maximal paths are studied in Section 4, where the main theorem is proven. Algorithms for minimizing a project-network are developed in Section 5. Potentially critical paths are investigated in Section 6. Section 7 gives some concluding remarks.

2. Problem Setting and Notations

Let a circuit-free digraph $G = (V, A)$ be given, where $V = \{0, 1, \dots, n\}$ is the set of vertices (activities) and A is the set of arcs. We assume that the numbering of the vertices V is correct in the sense that the inclusion $(i, j) \in A$ implies the inequality $i < j$. Let $P_{ij}(G)$ denote the set of all pathesν_{ij} in the digraph G started at vertex $i \in V$ and finished at vertex $j \in V$. The digraph G is called a project-network (or simply, a network) if for any vertex $i, 0 < i < n$, both sets $P_{0i}(G)$ and $P_{in}(G)$ are not empty and vertex 0 (vertex n) is the only vertex of the digraph G having a zero in-degree (a zero outdegree) called a source vertex (a sink vertex, respectively) in the network G. In what follows, only networks $G = (V, A)$ are considered.

The durations of the activities 0 and n are equal to zero. The duration of the activity $i, 0 < i < n$, may take any value from the given closed interval (segment) $[a_i, b_i]$, where $a_i \leq b_i$. For the source vertex 0 and the sink vertex n, we obtain $a_0 = b_0 = 0$ and $a_n = b_n = 0$, respectively. Let the weight of path $\nu_{rs} \in P_{rs}(G)$ be defined as the sum of the durations of all vertices (activities) included in the path ν_{rs}.

Since the duration of the activity $i \in V \setminus \{0, n\}$ may be equal to any value $t_i \in [a_i, b_i]$, the weight of the path $\nu_{rs} \in P_{rs}(G)$may remain uncertain until the completion of the project. As far as the weight of the path $\nu_{rs} \in P_{rs}(G)$ is uncertain at the stage of project planning, we denote the weight of the pathν_{rs} as $L^t(\nu_{rs})$ with a mandatory indication of the fixed vector t of the activity durations: $t = (t_0, t_1, \dots, t_n)$, where $a_i \leq t_i \leq b_i,\ i \in V$.

If $a_i = b_i$ for each activity $\in V$, then $G = (V, A)$ is a deterministic project-network, whose total duration is equal to the critical (maximal) weight $L^t(\nu_{0n})$ of the critical path (a path of maximal weight) $\nu_{0n} \in P_{0n}(G)$ from the source vertex 0 to the sink vertex n with the fixed vector $t = a = (a_0, a_1, \dots, a_n) = b = (b_0, b_1, \dots, b_n)$ of the activity durations.

On the other hand, if equality $a_i = b_i$ does not hold for at least one vertex (activity) $\in V$, i.e., $a_i < b_i$, then at the stage of project planning it is not clear which path $\nu_{0n} \in P_{0n}(G)$ has a maximal weight. Therefore, it is necessary to construct a specific set of paths in the digraph G belonging to the set $P_{0n}(G)$. Such a set of paths must contain at least one critical path for any vector t of the feasible durations. For the investigation of an uncertain project-network, we introduce the following dominance relation on the set $P_{ij}(G)$, which is reexive and transitive.

Definition 1 *The path* $\lambda_{ij} \in P_{ij}(G)$ *dominates the path* $\nu_{ij} \in P_{ij}(G)$, *if for any feasible real vector* $t = (t_0, t_1, \dots, t_n)$, $a_i \le t_i \le b_i$, *of the activity durations the inequality* $L^t(\lambda_{ij}) \ge L^t(\nu_{ij})$ *holds. A minimal (with respect to inclusion) set of paths* $H_{ij}(G) \subseteq P_{ij}(G)$ *is called a minimal dominant set for the ordered pair* $(i \to j)$ *of the vertices, if for any path* $\nu_{ij} \in P_{ij}(G)$, *there exits a path* $\lambda_{ij} \in H_{ij}(G)$ *dominating path* ν_{ij}. *A minimal dominant set for the ordered pair* $(0 \to n)$ *is called a minimal dominant set for the network* G.

If the equality $a_i = b_i$ holds for any activity $i \in V$, a minimal dominant set $H_{0n}(G)$ for the network G is determined by any critical path $\nu_{0j} \in P_{0j}(G)$ existing in the network G, i.e., $H_{0n}(G) = \{\nu_{0n}\}$. Indeed, any critical path in the deterministic network G dominates any path from the set $P_{0n}(G)$.

In Sections 3 - 5, we show how to construct a project-network G^* for the given project-network G such that the network G^* has the source vertex 0 and the sink vertex n and the equality $H_{0n}(G) = P_{0n}(G^*)$ holds.

In Section 6, we show how to construct a project-network G^0 for the given project-network G such that the network G^0 has the source vertex 0 and the sink vertex n and for any path $\lambda_{0n} \in P_{0n}(G^0)$, there exists a feasible vector t of the activity durations such that the path λ_{0n} is critical for this vector t.

3. Dominance criteria

Let the set of vertices belonging to path ν_{ij} be denoted as $[\nu_{ij}]$ and the set of arcs belonging to path ν_{ij}be denoted as $\{\nu_{ij}\}$. For simplicity, the indexes i and j may be omitted in the notation ν_{ij} of the path ν_{ij} , i.e., $\nu = \nu_{ij}$, if a misunderstanding does not arise. The following criterion has been proven in [3].

Lemma 1 *The path* λ *dominates the path* ν *if and only if*

$$\sum_{i \in [\lambda] \setminus [\nu]} a_i \ge \sum_{i \in [\lambda] \setminus [\nu]} b_i \tag{1}$$

It is easy to show that Lemma 1 implies the following one.

Lemma 2 *The path* λ *dominates the path* ν *if and only if*

$$L^{c(\lambda)}(\lambda) \ge L^{c(\lambda)}(\nu) \tag{2}$$

$$\textit{where } c(\lambda) = \big(c_0(\lambda), c_1(\lambda), \dots, c_n(\lambda)\big) \textit{ and } c_i(\lambda) = \begin{cases} a_i, i \in [\lambda], \\ b_i, i \notin [\lambda]. \end{cases}$$

Next, we prove the following

Lemma 3 *If the path* ν *does not dominate the path* λ*, then for any subset* $Z \subseteq [\nu] \cap [\lambda]$*, the following inequality holds:*

$$\sum_{i \in [\nu] \setminus Z} a_i < \sum_{i \in [\lambda] \setminus Z} b_i. \tag{3}$$

Proof: We consider an arbitrary subset $Z \subseteq [\nu] \cap [\lambda]$. The following equalities hold:

$$\sum_{i \in [\nu] \setminus Z} a_i = \sum_{i \in [\nu] \setminus [\lambda]} a_i + \sum_{i \in [\nu] \cap [\lambda] \setminus Z} a_i, \tag{4}$$

$$\sum_{i \in [\lambda] \setminus Z} b_i = \sum_{i \in [\lambda] \setminus [\nu]} b_i + \sum_{i \in [\lambda] \cap [\nu] \setminus Z} b_i, \tag{5}$$

Since $a_i \leq b_i$ for any vertex $i \in V$, we obtain

$$\sum_{i \in [\nu] \cap [\lambda] \setminus Z} a_i < \sum_{i \in [\nu] \cap [\lambda] \setminus Z} b_i. \tag{6}$$

Since the path ν does not dominate the path λ, due to Lemma 1, we obtain

$$\sum_{i \in [\nu] \setminus [\lambda]} a_i < \sum_{i \in [\lambda] \setminus [\nu]} b_i. \tag{7}$$

The relations (4) - (7) imply inequality (3). Lemma 3 has been proven.

4. An α-maximal path

For minimizing a project-network, we shall use the paths $\mu_{xy} \in P_{xy}(G)$,which have a maximal weight for the minimal values $(a_0, a_1, \dots, a_n) = a$ of the activity durations.

Definition 2 *The path* $\mu_{xy} \in P_{xy}(G)$ *is called* α *-maximal, if*

$$L^{\alpha}(\mu_{xy}) = \max_{\nu_{xy} \in P_{xy}(G)} L^{\alpha}(\nu_{xy}).$$

In what follows, the letter μ in the notation μ_{xy} will be used only for indicating an α-maximal path from the vertex x to the vertex y. If there are several such paths, then μ_{xy} indicates a path such that its value $\sum_{i \in [\mu_{xy}]} b_i$ is maximal. If there are also several such paths, then μ_{xy} indicates a path that has a minimal lexicographical order of its vertices. Thus, the path μ_{xy} is uniquely determined for each ordered pair $(x \rightarrow y)$ of the vertices with $x < y$.

4.1 Properties of a-maximal paths

The algorithms developed in Section 5 for minimizing a project-network G are based on choosing the α -maximal path μ_{xy}. For the proof of the main Theorem 1, we need three lemmas, which follow.

Lemma 4 *Assume that for the vertices* $x \in V$ *and* $y \in V$*, the* α *-maximal path* μ_{xy} *does not dominate another path from the set* $P_{xy}(G)$*. Then the set* $P_{xy}(G)$ *does not contain a path that dominates the path* μ.

Proof: We assume that the α-maximal path μ_{xy} does not dominate another path from the set $P_{xy}(G)$, however, there exists a path $\lambda_{xy} \in P(G)$, which is different from the path μ_{xy} and which dominates the path μ_{xy}.

Then, due to Lemma 1, we obtain

$$\sum_{i\in[\lambda_{xy}]\setminus[\mu_{xy}]} a_i \geq \sum_{i\in[\mu_{xy}]\setminus[\lambda_{xy}]} b_i \geq \sum_{i\in[\mu_{xy}]\setminus[\lambda_{xy}]} a_i . \tag{8}$$

Due to Definition 2 of the α-maximal path μ_{xy}, we obtain

$$\sum_{i\in[\lambda_{xy}]\setminus[\mu_{xy}]} a_i \leq \sum_{i\in[\mu_{xy}]\setminus[\lambda_{xy}]} a_i \leq \sum_{i\in[\mu_{xy}]\setminus[\lambda_{xy}]} b . \tag{9}$$

From relations (8) and (9), we obtain

$$\sum_{i\in[\lambda_{xy}]\setminus[\mu_{xy}]} a_i = \sum_{i\in[\mu_{xy}]\setminus[\lambda_{xy}]} b_i = \sum_{i\in[\mu_{xy}]\setminus[\lambda_{xy}]} a_i . \tag{10}$$

Since the path μ_{xy} does not dominate the path λ_{xy}, Lemma 1 implies

$$\sum_{i\in[\mu_{xy}]\setminus[\lambda_{xy}]} a_i < \sum_{i\in[\lambda_{xy}]\setminus[\mu_{xy}]} b_i . \tag{11}$$

From (10) and (11), we obtain

$$\sum_{i\in[\lambda_{xy}]\setminus[\mu_{xy}]} a_i < \sum_{i\in[\lambda_{xy}]\setminus[\mu_{xy}]} b_i. \tag{12}$$

The equalities (10) imply $L^a(\lambda_{xy}) = L^a(\mu_{xy})$. Thus, due to Definition 2 of the α -maximal pathμ_{xy}, we obtain

$$\sum_{i\in[\lambda_{xy}]\setminus[\mu_{xy}]} b_i \leq \sum_{i\in[\mu_{xy}]\setminus[\lambda_{xy}]} b_i. \tag{13}$$

The equalities (10) and the inequality (13) imply

$$\sum_{i\in[\lambda_{xy}]\setminus[\mu_{xy}]} b_i \leq \sum_{i\in[\lambda_{xy}]\setminus[\mu_{xy}]} a_i. \tag{14}$$

The obtained inequalities (12) and (14) contradict one to each other. Thus, our assumptions that the α-maximal path μ_{xy} does not dominate another path from the set $P_{xy}(G)$ but there exists a path $\lambda_{xy} \in P(G)$, which is different from the path μ_{xy} and which dominates the path μ_{xy}, are wrong. Lemma 4 has been proven.

Lemma 4 implies

Corollary 1 *If for any vertices* $x \in V$ *and* $y \in V$, *the* α*-maximal path* μ_{xy} *does not dominate any other path from the set* $P_{xy}(G)$, *then* $\mu_{xy} \in H(G)$.

We also need the following two lemmas.

Lemma 5 *If the path* λ_{0n} *dominates the path* ν_{0n} *in the network* G *and* $[\lambda_{0n}]\backslash[\nu_{0n}] = \{x_1, x_2, \dots, x_s\}$, *where* $x_1 < x_2 < \cdots < x_s$ *and* $s \geq 2$, *then there exists an index* $i\ (1 \leq i \leq s-1)$ *such that the path* $\lambda_{x_{i,}x_{i+1}}$ *dominates the path* $\nu_{x_{i,}x_{i+1}}$.

Proof: We assume that the path λ_{0n} dominates the path ν_{0n}, however, for any index $i\ (1 \leq i \leq s-1)$, the path $\lambda_{x_{i,}x_{i+1}}$ does not dominate the path $\nu_{x_{i,}x_{i+1}}$.

Then for any index i, due to Lemma 2, we obtain the inequality

$$L^{c(\lambda_{0n})}(\lambda_{x_{i,}x_{i+1}}) < L^{c(\lambda_{0n})}(\nu_{x_{i,}x_{i+1}}) \quad (15)$$

contradicting inequality (2). Note that $x_1 = 0, x_s = n$,

$$\lambda_{0n} = \cup_{i=1}^{s-1} \lambda_{x_{i,}x_{i+1}} \text{ and } \nu_{0n} = \cup_{i=1}^{s-1} \nu_{x_{i,}x_{i+1}}$$

Therefore, we obtain the following two equalities:

$$L^{c(\lambda_{0n})}(\lambda_{0n}) = \sum_{i=0}^{s-1} L^{c(\lambda_{0n})}(\lambda_{x_{i,}x_{i+1}}) \text{ and } L^{c(\lambda_{0n})}(\nu_{0n}) = \sum_{i=0}^{s-1} L^{c(\lambda_{0n})}(\nu_{x_{i,}x_{i+1}}).$$

Due to inequality (15), we obtain $L^{c(\lambda_{0n})}(\lambda_{0n}) < L^{c(\lambda_{0n})}(\nu_{0n})$. Hence, due to Lemma 2, the path λ_{0n} does not dominate the path ν_{0n}. This contradiction to our assumption completes the proof of Lemma 5.

Lemma 6 *For the given network* G, *the equality* $P_{0n}(G) = H_{0n}(G)$ *holds if and only if for any pair of vertices* $x \in V$ *and* $y \in V$, *the equality* $P_{xy}(G) = H_{xy}(G)$ *holds.*

Proof: Necessity. We assume that there exist vertices $x \in V$ and $y \in V$ such that $P_{xy}(G) \neq H_{xy}(G)$. Hence, there exist paths λ_{xy} and ν_{xy} in the set $P_{xy}(G)$ such that the path λ_{xy} dominates the path ν_{xy}. Then there exist a path $\lambda_{0n} = (\lambda_{0x}, \lambda_{xy}, \lambda_{yn})$ and a path $\nu_{0n} = (\lambda_{0x}, \nu_{xy}, \lambda_{yn})$ in the set $P_{0n}(G)$ such that the path λ_{0n} dominates the path ν_{0n}. As a result, $P_{0n}(G) \neq H_{0n}(G)$.

We obtained a contradiction.

Sufficiency. Since x and y may be arbitrary vertices in the network G, we can assume that $x = 0$ and $y = n$. As a result, the equality $P_{xy}(G) = H_{xy}(G)$ turns into the equality $P_{0n}(G) = H_{0n}(G)$. Lemma 6 has been proven.

4.2 Main theorem

Now, we can prove the main result as follows.

Theorem 1 *The equality* $P_{0n}(G) = H_{0n}(G)$ *holds for the digraph* G *if and only if for any pair of vertices* $i \in V$ *and* $j \in V, i < j$, *the* α*-maximal path* μ_{ij} *does not dominate any other path* $\nu_{ij} \in P_{ij}(G)$.

Proof: Necessity. Let there exist a pair of vertices i and j in the digraph G such that the a-maximal path μ_{ij} dominates some path $\nu_{ij} \in P_{ij}(G)$, $\mu_{ij} \neq \nu_{ij}$. Since the digraph G is a network, there exist a path $\nu_{0i} \in P_{0i}(G)$ and a path $\nu_{jn} \in P_{jn}(G)$. Therefore, there exists a path $(\nu_{0i}, \mu_{ij}, \nu_{jn}) \in P_{0n}(G)$ that dominates the path $(\nu_{0i}, \nu_{ij}, \nu_{jn}) \in P_{0n}(G)$. Therefore, $P_{0n}(G) \neq H_{0n}(G)$.

Sufficiency. Let the condition of Theorem 1 hold. We shall prove that for any pair of the vertices $x \in V$ and $y \in V, x < y$, no path from the set $P_{xy}(G)$ dominates another path from the set $P_{xy}(G)$.

<u>1.</u> If $|P(G)| \leq 1$, then the above claim holds.

<u>2.</u> Let $|P_{xy}(G)| = 2$, i.e., along with the path μ_{xy}, there exists a path $\nu_{xy} \in P_{xy}(G)$ such that $\mu_{xy} \neq \nu_{xy}$. Due to the condition of the theorem, the path μ_{xy} does not dominate the path ν_{xy}. Due to Lemma 4, the path ν_{xy}. does not dominate the path μ_{xy}.

<u>3.</u> Let $|P_{xy}(G)| = p + 1 > 2$ and $P_{xy}(G) = \{\mu_{xy}, \nu^1{}_{xy}, \ldots, \nu^p{}_{xy}\}$. Due to the condition of the theorem, the path μ_{xy} does not dominate any path from the set $\{\nu^1{}_{xy}, \ldots, \nu^p{}_{xy}\}$. Thus, due to Lemma 4, we obtain that no path from the set $\{\nu^1{}_{xy}, \ldots, \nu^p{}_{xy}\}$dominates the path μ_{xy}.

We assume that there exist a path $\emptyset_{xy}$ and a path ψ_{xy} in the set $\{\nu^1{}_{xy}, \nu^2{}_{xy}, \ldots, \nu^p{}_{xy}\}$such that the path $\emptyset_{xy}$ dominates the path ψ_{xy}. The following two cases are possible.

<u>3.1.</u> Case $[\emptyset_{xy}] \cap [\psi_{xy}] = \{x, y\}$.

Since $[\emptyset_{xy}] \backslash [\psi_{xy}] = [\emptyset_{xy}] \backslash \{x, y\}$ and $[\psi_{xy}] \backslash [\emptyset_{xy}] = [\psi_{xy}] \backslash \{x, y\}$, then assuming that the path $\emptyset_{xy}$ dominates the path ψ_{xy}, due to Lemma 1, we obtain

$$\sum_{k\in[\emptyset_{xy}]\backslash\{x,y\}} a_k \geq \sum_{k\in[\psi_{xy}]\backslash\{x,y\}} b_k \tag{16}$$

Definition 2 of the a-maximal path μ_{xy} implies

$$\sum_{k\in[\mu_{xy}]\backslash\{x,y\}} a_k \geq \sum_{k\in[\emptyset_{xy}]\backslash\{x,y\}} b_k \tag{17}$$

Since the path μ_{xy} does not dominate the path ψ_{xy}, Lemma 3 implies

$$\sum_{k\in[\mu_{xy}]\backslash\{x,y\}} a_k < \sum_{k\in[\psi_{xy}]\backslash\{x,y\}} b_k \tag{18}$$

Inequalities (16) - (18) imply the contradicting inequality:

$$\sum_{k\in[\mu_{xy}]\backslash\{x,y\}} a_k < \sum_{k\in[\mu_{xy}]\backslash\{x,y\}} a_k.$$

3.2. Case $[\emptyset_{xy}] \cap [\psi_{xy}] = \{x = x_1, x_2, \ldots, x_r = y\}$, $x_1 < x_2 < \ldots < x_r$, $r > 2$.

Since it is assumed that the path $\emptyset_{xy}$ dominates the path ψ_{xy}, due to Lemma 5, there exists an index $s\,(1 \leq s \leq r-1)$ such that the path $\emptyset_{x_s x_{s+1}}$ dominates the path $\psi_{x_s x_{s+1}}$.

Assume that between the vertices x_s and x_{s+1}, there are only two paths $\emptyset_{xy}$ and ψ_{xy}, i.e., $|P_{x_s x_{s+1}}(G)| = 2$. Then, arguing similarly as in the above point 2 for $x = x_s$ and $y = x_{s+1}$, we obtain a contradiction to the assumption that the path $\emptyset_{x_s x_{s+1}}$ dominates the path $\psi_{x_s x_{s+1}}$.

Assume that between the vertices x_s and x_{s+1}, there is an additional path, i.e., $|P_{x_s x_{s+1}}(G)| > 2$. Then, arguing similarly as in the point 3.1 for $x = x_s$ and $y = x_{s+1}$, we obtain a contradiction to the assumption that the path $\emptyset_{x_s x_{s+1}}$ dominates the path $\psi_{x_s x_{s+1}}$.

Thus, it has been proven that for any pair of vertices $x \in V$ and $y \in V$, $x < y$, the equality $P_{xy}(G) = H_{xy}(G)$ holds. Due to Lemma 6, this equality implies $P_{0n}(G) = H_{0n}(G)$. Theorem 1 has been proven.

5. Minimizing a project-network

The proven criteria for the path domination and Theorem 1 are used in two algorithms described in the following subsection.

5. 1. Algorithms for minimizing a project-network

Let V_i^- (V_i^+, respectively) denote the set of all direct predecessors (successors) of the vertex $i \in V$ in the digraph G. First, we describe Algorithm 1 for deleting arcs from the digraph G in order to destroy all paths $U_{ij}(G) \subseteq P_{ij}(G)$ that dominate the α-maximal path μ_{ij} .

At each iteration of Algorithm 1, a subgraph of the digraph G is considered, which is a network with the source vertex $i \in V$ and the sink vertex $n \in V$. For determining the vertices of such a subgraph, we shall mark the vertices of the set $\{i, i + 1, \ldots, n\} \subseteq V$ via considering them in increasing order of their numbers. In the description of Algorithm 1, W_i denotes the set of the marked vertices, $t^a{}_i$ and $t^b{}_i$ denote the earliest completion time of the activity i for the vector a and the vector b of the activity durations, respectively.

Algorithm 1

Step 1. Determine the ordered sets V_k^- and V_k^+ for each vertex $k \in V$. Mark vertex $i = 0$ and set $W_i := \{0\}$.

IF $|V_k^+| \geq 2$ **GOTO** step 13 **OTHERWISE GOTO** step 10.

Step 2. **IF** set V_j^- contains at least one non-marked vertex, i.e., $V_j^- \cap W_i \neq \emptyset$

GOTO step 4 **OTHERWISE GOTO** step 3.

Step 3. Set $j := j + 1$. **IF** $j \leq n$ **GOTO** step 2

OTHERWISE GOTO step 7.

Step 4. Calculate the earliest completion times $t^a{}_i$ and $t^b{}_i$ of the activity $j \in V$ with the vector a and the vector b of the activity durations, respectively:

$t^a{}_i = \max\{t^a{}_k + a_i : k \in V_j^- \cap W_i\}$ and $t^b{}_i = \max\{t^b{}_k + b_j : k \in V_j^- \cap W_i\}$.

Step 5. Determine the set of paths $P^*{}_{0k}(G) = \{\nu_{0k} : k \in V_j^- \cap W_i\}$

Step 6. For each path $\nu_{0k} \in P^*{}_{0k}(G)$ with the marked vertex $k \in V_j^- \cap$, test the inequality

$$t^a{}_j - a_j \geq L^b(\nu_{0k}). \quad (19)$$

IF inequality (19) holds **THEN** include the path (ν_{0k}, j) into the set $U_{0j}(G)$ of dominant paths.

Step 7. Determine the arc set $A^i \subset A$ such that for any path $(\nu_{0k,}\, j) \in U_{0j}(G)$, the condition $\{(\nu_{0k,}\, j)\} \cap A^i \neq \emptyset$ holds and for any path $(\nu_{0k,}\, j) \in {P^*}_{0j}(G) \backslash U_{0j}(G)$, the equality $\{(\nu_{0k,}\, j)\} \cap A^i = \emptyset$ holds.

Step 8. Delete the arc set A^i from the digraph $G_i = (V, A_i)$.

Step 9. Mark the vertex j, i.e., set $W_i := W_i \cup \{j\}$. Set $j := j + 1$.

IF $j \leq n$ **GOTO** step 2 **OTHERWISE GOTO** step 10.

Step 10. Set $i := i + 1$. Assume that all vertices V are not marked, i.e., set

$W_i := \emptyset$. **IF** $i \geq n - 1$ **STOP**.

OTHERWISE GOTO step 11.

Step 11. Let $G_i = (V, A_i)$ denote the digraph obtained from the digraph G_{i-1} via deleting the set A^i of arcs in step 8.

IF step 8 was not realized yet or the arcs A^i were not deleted yet **THEN** set $G_i \coloneqq G_{i-1}$.

Step 12. Mark vertex i, i.e., set $W_i := \{i\}$.

IF $|{V^+}_i| \geq 2$ **GOTO** step 13 **OTHERWISE GOTO** step 10.

Step 13. Set $G_i = (V, A_i) \coloneqq G$, ${t^a}_i = {t^b}_i := 0, j := i + 1$, and $i := 0$.

GOTO step 2.

Implementing Algorithm 1 to the digraph G destroys all paths starting from vertex i and ending in vertex j which dominate the α-maximal path μ_{ij} (see inequality (19)). However, after deleting the set A^i of arcs, the resulting digraph may contain arcs and vertices, which are not contained in any path from the source vertex 0 to the sink vertex n. To delete such arcs and vertices, one can use the following algorithm.

Algorithm 2

Step 1*. Choose an arbitrary arc $(0, i)$ in the digraph $\acute{G} = (\acute{V}, \acute{A})$ obtained from the digraph G after implementing Algorithm 1. Mark this arc $(0, i)$ and pass to vertex i in order to consider the vertex i.

Step 2*. Assume that after a sequence of previous steps, one passed to vertex i in order to consider this vertex.

Step 3*. **IF** in the resulting digraph, there exist non-marked arcs starting from vertex i **THEN** choose any such non-marked arc, say arc (i, j), mark arc (i, j), pass to vertex j **GOTO** step 2* and realize step 2* with setting $i := j$.

IF vertex i is not the sink vertex n or there is no arc starting from vertex i **THEN** delete the arc, which was used when passing to vertex i. **IF** the deleted arc was a single arc ending in vertex i, delete the vertex i as well. Pass to the previous vertex, say vertex k, GOTO step 2* and realize step 2* with setting $i := k$.

IF vertex i is the sink vertex n **THEN** pass to the previous vertex k **GOTO** step 2* and realize step 2* with setting $i := k$.

IF vertex i is the source vertex 0 and all arcs starting from i are marked **THEN GOTO** step 4*.

IF vertex i is not the sink vertex n and all arcs starting from i are marked **THEN** pass to the previous vertex k, GOTO step 2* and realize step 2* with setting $i := k$.

Step 4*. **IF** the resulting digraph contains non-considered vertices **THEN** delete them. In both cases **STOP**.

It is easy to convince that the condition of Theorem 1 holds for the digraph $G^* = (V^*, A^*)$ obtained from the digraph $G = (V, A)$ after implementing first Algorithm 1 and then Algorithm 2.

5.2. Complexity of Algorithms 1 and 2

In what follows, the notations $m = |A|$ and m^* are used, where m^* denotes the number of arcs in the digraph G, which are deleted due to the implementation of Algorithm 1.

Lemma 7 *Algorithm 1 can be realized in* $O(nm)$ *time.*

Proof: In step 1, the ordered sets $V^-{}_k$ and $V^+{}_k$ may be constructed after a single consideration of the arcs of the set A which takes $O(m)$ time. Realizing steps 2 - 6 requires two considerations of the arcs of the set A, which take $O(m)$ time. Testing the subgraphs G_i of the digraph G can be realized in steps 7 and 8, the number of subgraphs G_i being restricted by n. Thus, the asymptotic complexity of Algorithm 1 is $O(nm)$.

Lemma 8 *Algorithm 2 can be realized in* $O(m)$ *time.*

Proof: In steps 1* - 3*, a pass through each arc $(i, j) \in A^*$ is realized no more than twice (in the direct direction of the arc and in the opposite direction of the arc). Step 4* is realized by a single test of all arcs $(i, j) \in A^*$. Thus, the complexity of Algorithm 2 can be restricted by $O(m - m^*)$ or moreroughly by $O(m)$.

Due to Lemmas 7 and 8, we can conclude that the implementation of Algorithm 1 and Algorithm 2 to the digraph G allow us to construct a minimal network G^* in $O(nm)$ time.

6. Potentially critical paths

Since the digraph G^* contains at least one critical path for any feasible activity durations $t = (t_0, t_1, \dots, t_n)$, $t_i \in [a_i, b_i]$, the digraph G^* allows us to determine the total duration of the project defined by the project-network G for any possible vector t of the activity durations. As it was mentioned in [2], it is also important for the project control to know the set of all paths in the project-network G, which may become critical for at least one feasible vector t of the activity durations.

Definition 3 *The path* $\lambda_{0n} \in P_{0n}(G)$ *is called potentially critical for the project-network* G*, if there exists a vector* $t = (t_0, t_1, \dots, t_n)$, $a_i \le t_i \le b$*, of the activity durations, for which the path* λ_{0n} *is critical, i.e., the equality*

$$L^t(\lambda_{0n}) = \max_{\nu_{0n} \in P_{0n}(G)} L^t(\nu_{0n})$$

holds.

Let $K_{0n}(G)$ denote the set of all potentially critical paths in the project-network G. Let the subgraph G^0 of the digraph G be also a network with the source vertex 0 and the sink vertex n such that $K_{0n}(G) = P_{0n}(G^0)$.

The knowledge of the set $K_{0n}(G)$ allows a manager of the project G to control effectively the realization of the project G. In fact, the set $K_{0n}(G)$ plays the same role in the uncertain project control as the set of all critical paths in the control of the deterministic project. Constructing the network G^0 may be realized similarly as constructing the network G^*, if instead of the dominance relation (see Definition 1) we shall use the following strong dominance relation, which is non-reflective and transitive.

Definition 4 *The path* $\lambda_{ij} \in P_{ij}(G)$ *strongly dominates the path* $\nu_{ij} \in P_{ij}(G)$*, if for any feasible vector* x *of the activity durations* $x = (x_0, x_1, \dots, x_n)$, $a_i \le x_i \le b_i$, *the inequality* $L^x(\lambda_{ij}) > L^x(\nu_{ij})$ *holds.*

It is easy to convince that Lemma 1 and Lemma 2 become correct for the strong dominance relation if the sign $\ge$ of the non-strict inequality is replaced by the sign $>$ of the strict inequality. Furthermore, analogues of Lemmas 3 - 6 and Theorem 1 may be proven for the project-network G^0 and for the strong dominance relation given on the set of paths in the project-network G. As a result, for constructing the network G^0 from the given network G, one can use Algorithm 1, in which the non-strict inequality (19) is replaced by the following strict inequality:

$$t^a{}_j - a_j > L^b(\nu_{0k}) \tag{20}$$

Then by implementing Algorithm 2 to the obtained network, the desired project-network G^0 will be constructed. Thus, constructing the project-network G^0 from the project-network G takes $O(nm)$ time.

In conclusion, we prove that a path containing a transitive arc $(x, y) \in A$ with the inequality $a_i > 0$ cannot be critical for any feasible vector t of the activity durations. An arc $(x, y) \in A$ is called transitive if there exists a path λ_{xy} starting from the vertex x and ending in the vertex y with a length more than one. It is clear that the above path λ_{xy} cannot include the arc (x, y).

Theorem 2 *If the path* $\lambda_{0n} \in P_{0n}(G)$ *contains at least one transitive arc* $(x, y) \in A$ *with* $a_i > 0$, *then* $\lambda_{0n} \notin K_{0n}(G)$ *and* $\lambda_{0n} \notin H_{0n}(G)$.

Proof: We consider a path $\lambda_{0n} = (\lambda_{0x}, (x, y), \lambda_{yn})$, where the arc (x, y) is transitive and the inequality $a_i > 0$ holds. Since the arc (x, y) is transitive, there exists a path λ_{xy} in the digraph G, which does not contain the arc (x, y). Therefore, there exists a path $\lambda^*{}_{0n} = (\lambda_{0x}, \lambda_{xy}, \lambda_{yn})$, which does not include the arc (x, y). Since the equality $[\lambda_{0n}] \setminus [\lambda^*{}_{0n}] = \emptyset$ holds, we obtain

$$\sum_{k \in [\lambda_{0n}] \setminus [\lambda^*{}_{0n}]} b_k = 0.$$

The inequality

$$\sum_{k \in [\lambda^*{}_{0n}] \setminus [\lambda_{0n}]} a_k > 0$$

implies

$$\sum_{k \in [\lambda^*{}_{0n}] \setminus [\lambda_{0n}]} a_k > \sum_{k \in [\lambda_{0n}] \setminus [\lambda^*{}_{0n}]} b_k$$

Due to Lemma 1, the path $\lambda^*{}_{0n}$ dominates the path λ_{0n}. Hence $\lambda_{0n} \notin K_{0n}(G)$. Due to the analogue of Lemma 1 for the strong domination on the set of paths, one can conclude that the path $\lambda^*{}_{0n}$ strongly dominates the path λ_{0n}. Hence, $\lambda_{0n} \notin H_{0n}(G)$. Theorem 2 has been proven.

Theorem 2 implies the following

Corollary 2 *If* $a_i > 0$, $0 < i < n$, *and the path* $\lambda_{0n} \in P_{0n}(G)$ *contains at least one transitive arc* $(x, y) \in A$, *then* $\lambda_{0n} \notin K_{0n}(G)$ *and* $\lambda_{0n} \notin H_{0n}(G)$.

For any feasible vector $t = (t_0, t_1, \dots, t_n)$, $t_i \in [a_i, b_i]$, $i \in V$, of the activity durations, the constructed network G^0 contains all potentially critical paths of the original network G. Therefore, in the control of the project-network G, it is sufficient to control only the network G^0 which is usually simpler than the original network G.

7. Concluding remarks

The above results proven in Sections 4 - 6 and the algorithms developed in Section 5 and Section 6 allow a manager to simplify the original project-network without loss of the main characteristics of the project-network. Indeed, the network G^* allows a manager to calculate the total duration of the project for any feasible activity durations. Furthermore, the network G^0 allows a manager to determine all possible critical paths, which may appear during a realization of the project.

For calculating different time reserves of the activity realization in the uncertain project-network, one can use calculations based on fuzzy logic. Unfortunately, such calculations are often time-consuming. The usage of the simplified network instead of the original one will simplify such calculations.

The developed approach to minimize the digraph without loss of the main digraph characteristics may be used for solving appropriate scheduling problems with uncertain (interval) input parameters [3, 4].

References

[1] Brucker, P., Drexl, A., Mohring, R., Neumann, K., and Pesch, E., 1999, Resource-constrained project scheduling: Notation, classification, models, and methods, European Journal of Operational Research, **112, 1**, 3 - 41.

[2] Chen, B., Pinar, M. C., Tansel, B., and Yaman H., 1998, Project management with interval activity durations. Proceedings of the Sixth International Workshop on Project Management and Scheduling, Istanbul, Turkey, 24 -26.

[3] Lai, T.-C. and Sotskov, Y. N., 1999, Sequencing with uncertain numerical data for makespan minimization. Journal of the Operational Research Society, **50**, 230 - 243.

[4] Lai, T.-C., Sotskov, Y. N., Sotskova, N. Y., and Werner, F., 1997, Optimal makespan scheduling with given bounds of processing times. Mathematical and Computer Modelling, **26, 3**, 67 - 86.

[5] Lootsma, F. A., 1989, Stochastic and fuzzy PERT. European Journal of Operational Research, **43**, 174 - 183.

[6] McCahon, C. S., 1993, Using PERT as an approximation of fuzzy project-network analysis. IEEE Transaction on Engineering Management, **40, 2**, 146 - 153.

[7] Tanaev, V. S., Sotskov, Y. N. and Strusevich, V. A., 1994, Scheduling Theory: Multi-Stage Systems. Kluwer Academic Publishers, Dordrecht, The Netherlands.

[8] Thulasiraman, K. and Swamy, M. N. S., 1992, Graphs: Theory and Algorithms. John Wiley & Sons, USA.