THE METHOD FOR IMPROVING STABILITY OF CONSTRUCTION PROJECT SCHEDULES THROUGH BUFFER ALLOCATION

Piotr Jaśkowski¹, Sławomir Biruk²

Lublin University of Technology, Faculty of Civil Engineering and Architecture,
Nadbystrzycka str. 40, 20-618 Lublin, Poland
E-mails: ¹p.jaskowski@pollub.pl (corresponding author); ²s.biruk@pollub.pl

Received 25 August 2010; accepted 28 March 2011

Abstract. The actual completion time of construction projects is rarely in accordance with initial plans. This is due to unforeseen events that affect the project execution. In the paper, the authors propose a method of improving the construction schedule reliability. The method is based on the idea of buffers allocation, whose proper sizing helps to reduce negative effects of random conditions on the progress and efficiency of project activities. The authors adopt a proactive approach to the construction of robust schedules to cope with multiple disruptions during project execution, applicable to two processes starting policies (as soon as possible or no sooner than on a predefined date). The proposed method, based on simulation technique and mathematical programming, was illustrated by an example. The results obtained by means of the proposed method were compared, in terms of schedule stability, to those of the float factor heuristic and starting time criticality heuristic procedures. The method allowed reducing cost of deviations from the planned activity start dates in random conditions.

Keywords: construction project scheduling, stable solution, robust schedule, buffering, risk management, project management, mixed linear-integer programming.

Reference to this paper should be made as follows: Jaśkowski, P.; Biruk, S. 2011. The method for improving stability of construction project schedules through buffer allocation, Technological and Economic Development of Economy 17(3): 429–444.

JEL Classification: M11, C61, C63.

1. Introduction

A schedule is the basic tool used in construction project management. In the case of projects that involve non-cyclic, non-uniform and non-rhythmical processes, the most popular planning techniques are network methods. Among them, the critical path method is used most often,
though it does not allow the planner to account for all constraints that occur in the practice of construction operations, in terms of both market conditions and technical operations.

Over the past few years, research efforts in the field of construction project planning concentrated on exact and heuristic procedures for generating project schedules, usually under assumption of deterministic conditions. Most of them considered a variety of constraints and optimization objectives (e.g. Jaskowski, Biruk 2009; Kaplinski 2008a, 2008b; Jaskowski, Sobotka 2006; Christodoulou 2009; Liu, Wang 2008).

Nevertheless, construction processes are especially prone to risk, which affects project performance (Kaplinski, Janusz 2006; Karlowski, Paslawski 2008; Enshassi et al. 2009). Uncertainty has become one of the major factors affecting project execution and ultimate success (Ben-Haim, Laufer 1998). The risk in construction business is very high (Zavadskas et al. 2010). Risk comes from many sources during the whole project life cycle (Shevchenko et al. 2008; Manik et al. 2008). Therefore, the risk assessment problem (Zavadskas et al. 2010; Zhao, Jiang 2009; Miao et al. 2009), the problem of decision-making in risk environment (Peldschus 2008; Kahraman, Kaya 2010; Kapliński, Tamošaitytė 2010; Ginevičius, Zubrecovas 2009), and risk handling and allocation (e.g. Perera et al. 2009; Kheirkhah et al. 2009) attract the interest of numerous researchers and practitioners. Capturing the risk in the form of risk indices is reckoned to help the planner to shape judgments and to make decisions.

A contractor’s project program expressed in the form of a schedule, regardless of the project development phase (bidding, organizing, managing on-site operations), is to be reliable in terms of not only the total project makespan, but also timing of particular tasks and activities – related with resource management. As a result of random events and conditions, a schedule that has been optimized for a set of particular objectives, but determined without consideration to disturbances that may occur, is likely to yield results significantly poorer than expected.

Project planning for uncertainty and risk has been the subject of numerous research efforts since introduction of the PERT model (e.g. Biruk, Jaskowski 2008; Blaszczyszk, Nowak 2009; Kaplinski 2008a; Nassar, Casavant 2008). The existing methods that express activity durations as random values (e.g. PERT, GANT, CYCLONE, Petri nets) and simulation models (e.g. Klimov, Merkuryev 2008; Kaplinski 2009) focus mainly on estimating either the probability of conforming to the contractual completion date, or the project duration at a predefined confidence level. The problem of practical workflow planning is thus somehow neglected, as the methods mentioned above do not account for the problem of defining activity starts (other than “early starts”) and introducing subcontractors (or resource management in general). Moreover, the analysis of schedules obtained by means of the above methods provides information in the form of probabilistic values, which is not easy to interpret for practical planning of the project logistics.

Deficiencies of the existing methods gave rise to the worldwide search for more reliable methods of project planning, such as predictive (or proactive) scheduling that is expected to provide robust schedules (immune to disturbances), thus counteracting instability and “nervousness” of a project plan.

It is believed that a perfect schedule should combine solution robustness (i.e. it should be stable) and quality robustness (i.e. it should be makespan protective) (Van de Vonder et
al. 2005). Therefore, recent analyses aim at constructing robust baseline schedules. They are often proactive, which seems more effective than adoption of a purely reactive approach of rescheduling and updating the plan during execution. Paslawski (2008) proposed to improve the efficiency of the reactive approach by increasing flexibility of construction process engineering, based on creating multiple variants for process realization options, thus enabling adaptation to current realization conditions.

A construction project is characterized by a high number of project participants and multitude of contract relations (Schieg 2008). The main cause of the need for stable baseline schedules is the “advance booking” of key staff or equipment (to guarantee their availability) and fixed delivery dates required by suppliers or subcontractors (Herroelen, Leus 2004). Scheduling with fixed dates of employing resources makes planning easier – allows the contractor to manage project portfolio and restrict propagation of interferences from one project to the other. Furthermore, not keeping to the contract due date and milestones is usually connected with penalties.

A widely used method to increase stability of a schedule with a predefined due date is including time buffers between processes (Jaskowski, Biruk 2010).

2. Literature review

Buffering is a common practice in project management. Lill (2009) recommended buffer in form of buildings with no duration limits in construction programme as labour management compensators for improving the arrangement of labour resources. Buffers in form of idle periods guarantee the completion of an activity or a project on time. However, the existing methods of time buffer positioning and sizing are often criticized by the researchers.

Goldratt (1997) introduced three types of time buffers according to their function. The project buffer is placed at the end of the critical chain (the longest chain of precedence of resources dependent processes) to protect the project due date. Feeding buffers are inserted to protect the critical chain against disruptions from non-critical processes and placed at points where non-critical chains join the critical chain. Resource buffers alongside the critical chain should ensure resources availability. Critical chain scheduling, or buffer management (CC/BM), being the Goldratt’s Theory of Constraints application to project management, assumes that the baseline schedule should be created using median durations estimates of the processes. However, introduction of the project buffer may lead to unnecessarily late project due dates, and the feeding buffers fail to prevent propagation of schedule disruptions in most cases (Van de Vonder et al. 2005). Moreover, CC/BM approach does not provide the baseline schedule with fixed start times of processes. Instead, processes are to start as soon as possible.

In contrast to the above approach, the authors assume that, as the project runs, some or all processes are not allowed to start before their start time planned in the baseline schedule. This policy is called railway scheduling.

Some planners treat a contingency buffer as an integral part of each activity, without a clear distinction from the original duration estimate (Park, Peña-Mora 2004). This approach and positioning a buffer at the end of an activity may lead to a so called “student syndrome” – a situation when a contractor responsible for a particular task waits until the last possible
moment to start. Thus, a buffer does not protect an activity start date causing unnecessary interruptions in work flow. Therefore, some authors suggest placing time buffers in front of processes (Goldratt 1997; Park, Peña-Mora 2004; Herroelen, Leus 2004; Van de Vonder et al. 2008).

The size of a contingency buffer is normally decided on the basis of individual experience and assigned in a uniform way, usually as a fixed fraction of activity duration, instead of considering particular qualities of each individual activity (Park, Peña-Mora 2004). Two main factors should be taken into consideration while determining the size of the buffer in front of a process:

- the variability of all the processes that precede the process in the schedule (because it affects the probability that process can start at its scheduled starting time),
- and the weight or unit cost of process start time disruption.

The evaluation of quality of a buffered schedule is generally based on measures which are used to quantify the capacity of this schedule to remain efficient in random circumstances (Ghezail et al. 2009), but there are no generally agreed definitions or measures of robustness. Al-Fawzan, Haouari (2005) proposed to measure the schedule robustness with the total sum of free floats. This way, the robustness of the schedule was defined as the ability to execute the project without any delays in start of project’s processes. Kobylanski, Kuchta (2007) state that a better measure of the schedule robustness would be the minimal free float. The planner’s task should be thus maximizing the minimum of free floats. This way, start date of each activity would be protected (which means solution robustness) and the makespan of the whole project would be assured (which means quality robustness). This approach fuses solution and quality robustness. There is also no need to determine weights of makespan and robustness criteria like in a bi-objective model presented by Al-Fawzan, Haouari (2005). Robustness of a schedule can be measured also by the minimum of the free float to duration ratios calculated for all processes (Kobylanski, Kuchta 2007).

According to Herroelen, Leus (2004) and Van de Vonder et al. (2005 and 2006), a stable schedule with acceptable makespan performance should minimize the instability cost function, defined as the weighted sum of the expected absolute deviations between the predicted start times and the value that the random variable of start time will assume during schedule execution.

An optimization model of the buffer sizing problem (minimizing the instability cost), concerning the case of disruption to a single activity, with discrete disruption scenario, was presented by Herroelen, Leus (2004). However, in terms of computational effort, it is inefficient to search for exact optimal solutions of practical complex problems. Therefore, the literature on the subject proposes numerous heuristic and meta-heuristic algorithms for allocating buffers throughout the schedule but only adequate for the railway policy for starting all of the processes.

The adapted float factor and resource flow dependent float factor heuristic (Van de Vonder et al. 2005, 2006; Herroelen, Leus 2004) relies on the activity weights, but it does not exploit the information on the activity duration distributions for making its buffering decisions.

The concept of a float factor was introduced by Tavares et al. (1998) as a way of augmenting the earliest processes start time by the same fraction of the total float. This is to reduce the risk of not meeting project due date and to decrease the discounted project cost.
This procedure starts with an unbuffered schedule and modifies it by adding safety buffers in the front of processes. The start time of a process $j$ is calculated as $s_j = s_j(U) + \alpha_j \cdot \text{float}_j$, where $s_j(U)$ denotes the earliest start of process $j$ in the initial unbuffered schedule, $\alpha_j$ – the float factor, and $\text{float}_j$ – the total float of process $j$ in the unbuffered schedule. The float factors are calculated as $\alpha_j = \beta_j / (\beta_j + \lambda_j)$, where $\beta_j$ is the sum of the weight of activity $j$ and the weights of all its transitive predecessors in the network, while $\lambda_j$ is the sum of the weights of all transitive successors of activity $j$. The weights of activities that start at time 0 are not included in the above sums as it is assumed that these activities can always start at their planned start dates. Thus, there is no need of buffering to cope with possible disruptions of their predecessors.

The virtual activity duration extension heuristic (Van de Vonder et al. 2005, 2006; Herroelen, Leus 2004) relies on the standard deviations of processes’ durations, used to compute a modified duration for constructing the baseline schedule. The standard deviations are used at each iteration of the algorithm to select a non-dummy process for which virtual duration extensions are made. Duration extensions require schedule (early starting times) update and allow the planner to calculate buffer size in the baseline schedule with original expected activity durations. At every iteration step, the instability cost of the schedule is evaluated by simulation.

The starting time criticality heuristic (Van de Vonder et al. 2005, 2006; Herroelen, Leus 2004) tries to combine information on activity weights and activity duration variances. Based on an initial unbuffered schedule, intermediate schedules are created, in iterative way, by adding a one-unit time buffer in the front of the most critical activity until increasing contingency buffers no longer improves stability. The process criticality is measured by a multiplication of the probability of a process’ not starting on its scheduled date and the unit cost of the process’ deviating from its start time. This probability is estimated under assumption that only one activity at a time disturbs the start time of the process.

An extensive simulation-based analysis of the performance of heuristic and meta-heuristic algorithms of buffer allocation presented in (Van de Vonder et al. 2008) revealed that heuristics that use information on the unit cost of a process’ start time deviation and process duration variability yield better solutions. Van de Vonder et al. (2008) present an improvement to the above heuristics based on searching better neighborhood solutions (their fitness is assessed by simulation technique), constructed by changing processes buffers sizes (in taboo search procedure) or starting times within displacement interval without affecting the starting times of other processes in the buffered or unbuffered schedule. This procedure reduces the instability cost by reallocating buffers and reducing floats. However, because of computational effort, it is adequate only for small test problems.

3. Proposed approach to buffer sizing

The precedence relationships between schedule activities, i.e. construction processes, have been modeled by a unigraph $G = (V, E)$, directed, acyclic, in activity-on-the-node representation with single start and end nodes. $V = \{1, 2, \ldots, n\}$ is a set of construction processes (nodes), $E \subset V \times V$ is a bi-argument relation representing precedence relationships between processes (arcs).
Each process is assigned a duration $d_j$, established using mean productivity estimates, equal to expected values of durations distributions. These duration estimates are used for creating the baseline schedule.

Each non-dummy process $j$ has a weight $c_j$ that denotes the unit cost of delaying its start, the delay being the difference between the actual and the start time assumed in the baseline schedule (nonnegative cost per unit time overrun on the start time of activity). A small activity weight reflects high scheduling flexibility or low instability cost. For example activities that depend on resources with non availability limits will be given a small weight. A heavy weight reflects small scheduling flexibility: deviations between actual and planned starting times are deemed very costly for the contractor, e.g. high penalties that are incurred when individual milestones or the project due date are not met (Herroelen, Leus 2004). The cost of the final dummy activity $n$ denotes the cost of delaying the project completion beyond the predetermined (contractual) due date $T_d$. As the contract gave high importance to meeting the planned due date of the project, a large weight had to be given to the activity marking the project completion.

Processes whose $c_j > 0$ start in accordance with railway policy – not before the start date predefined in the baseline schedule. This policy can be applied also to the remaining processes.

The objective is to build a stable precedence feasible schedule with acceptable makespan performance by minimizing the function of schedule instability cost:

$$C = \sum_{j=1}^{n} c_j E(s_j - s^\prime_j),$$

defined as the weighted sum of the expected shifts of the actual starting time $s_j$ from the predicted starting time $s^\prime_j$ of the baseline schedule.

Buffers $\delta_j$ are introduced to prevent propagation of disruptions throughout the schedule. These buffers are idle periods (gaps) in the schedule between the planned start of a process considered and the latest planned finish of its predecessors (the earliest start of the process). The buffers can absorb disruptions in process durations without affecting other processes and the schedule logic. This way, coordination of resource and material procurement for each activity can be performed as smoothly as possible.

The buffer size should stay – in the opinion of the authors – in proportion to the process variability and to the cost of a possible delay beyond the earliest start in the baseline schedule.

To establish the possible delay it is necessary to perform simulations, especially in the case of different process starting policies used during project execution, such as the railway or as early as possible policy.

Buffer size is established in the following steps:

1. Calculation of the minimal project duration $T_{\min}$, the early start times of all activities, $s^0_j$, and their total floats, $\text{float}^0_j$, according to the baseline schedule based on expected values of activity durations, $d_j$.

2. Simulations of the project progress on the basis of the network model assuming that activity durations are random variables of predefined probability distribution, and that the activities start in accordance with the assumed starting policy.

The simulation enables the planner to estimate expected values of activity delays $\Delta s_j = s^\prime_j - s^0_j$, where $s^\prime_j$ is the mean start date of a process $j \in V$ calculated on the basis of simulation results.
Assuming that the contractual project duration is known and equals $T_d$, the total float of an activity $j \in V$ is $\text{float}_j = \text{float}_j^0 + T_d - \text{min}_j$.

The existing floats in the baseline schedule, having the form of time buffers, should be allocated among processes according to process weights, $w_j$, calculated according to the formula:

$$w_j = c_j \cdot \left(\Delta s_j + 3\sigma_j\right),$$

where $\sigma_j$ is the standard deviation of the simulated activity start dates.

This approach to determining weights is similar to the process criticality concept presented by Kuchta (2000) and results from the fact that buffer sizes less than $\Delta s_j + 3\sigma_j$ do not eliminate propagation of disruption in the network.

3. Buffer sizes calculations. The mathematical model of the buffer $\delta_j$ sizing problem can be formulated as follows:

$$\max z = \min{\left\{ \frac{\delta_j}{\text{float}_j w_j} \right\}},$$

$$s_1 = 0,$$

$$s_j - \delta_j \geq s_i + d_i, \; \forall (i, j) \in E,$$

$$s_n + d_n \leq T_d,$$

$$s_j \geq 0, \; \forall j \in V,$$

$$\delta_j \geq 0, \; \forall j \in V,$$

$$\delta_j = 0, \; \forall j \in V \setminus H,$$

$$\delta_j \in \text{int}, \; \forall j \in H,$$

where $H = \{ j : w_j > 0 \}$.

The objective function (3) is used for proper buffer sizing (float allocation) according to processes’ weights. Execution of the first process of the project (i.e. a process that has no predecessors) starts at the moment 0 (4). The condition (5) is required to establish processes’ start dates. The project should end before the predefined deterministic due date $T_d$ (6).

To facilitate the computations a mixed linear-integer model can be constructed by modifying the objective function and adding one condition:

$$\max z = \xi,$$

$$\frac{\delta_j}{\text{float}_j} \geq w_j \xi, \; \forall j \in H.$$
set of all solutions. Thus, buffers of non-critical activities may be increased or decreased unless it raises the cost of schedule instability.

It is possible to improve the solution using the following procedure that should be run for non-critical activities of the same path between the project’s start and end:

1. Find value of the schedule instability cost, \( C(1) \), on the basis of simulation using current schedule.

2. For each non-critical activity in the same path, find values \( C'_j^+, C'_j^- \) of instability cost function (1) on the basis of simulations of new schedules created by increasing or decreasing the buffer by a unit of time.

3. If there exists an activity whose \( \Delta C_j^+ = C'_j^+ - C < 0 \), go to step 4. In other case, finish the procedure and consider the current schedule to be the best.

4. Establish the number of the activity \( j^* \) for which \( \Delta C_j^+ \) or \( \Delta C_j^- \) is minimal. Then change its buffer by a unit of time so to get maximum reduction of instability cost.

5. The resulting schedule is considered the current solution. Assume \( C := C'_j^{*\pm} \) and go to step 2.

In the remainder of the paper, the above procedure is referred to as the improvement procedure.

4. Example

Figure 1 presents an example of a construction project network model. Table 1 lists the estimates of parameters of activity durations under assumption that probability distributions of these durations are triangular (\( a_j \) – the shortest possible duration, \( m_j \) – the most probable duration, \( b_j \) – the longest possible duration, \( d_j \) – the expected duration), and \( c_j \) – the unit costs of delaying the activities’ start. Minimal project duration calculated on the basis of mean process durations is 75 days. The proposed method was used to find the project schedule of due date 80 days after start that fulfills a requirement of minimizing the expected cost of delaying the start of processes beyond their predicted (scheduled) start times.

The expected delay of each process and standard deviations was established by Monte Carlo simulation (30 000 runs). Simulations were conducted by means of GPSS World\textsuperscript{tm} Personal Version by Minuteman Software (education license) for two variants of the project execution:

- in the first variant, all activities start according to the railway policy,
- in the second variant, only the activities \( j \in H \) start no sooner than on the dates \( s_j^0 \), defined in the baseline schedule, and the remaining activities start immediately after all their predecessors finish.

![Fig. 1. Precedence relationships among processes of the example](image)
Table 1. Estimates of processes durations and costs of start delaying (example)

<table>
<thead>
<tr>
<th>$j$</th>
<th>Process</th>
<th>$a_j$</th>
<th>$m_j$</th>
<th>$b_j$</th>
<th>$d_j$</th>
<th>$c_j$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>start</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>site preparation</td>
<td>5</td>
<td>6</td>
<td>8</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>excavations</td>
<td>4</td>
<td>5</td>
<td>7</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>foundation</td>
<td>8</td>
<td>10</td>
<td>15</td>
<td>11</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>basement walls</td>
<td>3</td>
<td>4</td>
<td>7</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>basement floor slab</td>
<td>3</td>
<td>4</td>
<td>7</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>brickwork</td>
<td>15</td>
<td>18</td>
<td>25</td>
<td>19</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>roof structure</td>
<td>7</td>
<td>8</td>
<td>10</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>roof cladding</td>
<td>5</td>
<td>6</td>
<td>8</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>10</td>
<td>external services</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>backfill</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>12</td>
<td>partition walls</td>
<td>3</td>
<td>5</td>
<td>8</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>plastering</td>
<td>8</td>
<td>10</td>
<td>15</td>
<td>11</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>painting</td>
<td>5</td>
<td>6</td>
<td>9</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>facade works</td>
<td>10</td>
<td>12</td>
<td>18</td>
<td>13</td>
<td>9</td>
</tr>
<tr>
<td>16</td>
<td>plumbing</td>
<td>10</td>
<td>12</td>
<td>15</td>
<td>12</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td>site works</td>
<td>14</td>
<td>16</td>
<td>22</td>
<td>17</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>finish</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
</tr>
</tbody>
</table>

Table 2. Results of calculations by the proposed method – the first variant

<table>
<thead>
<tr>
<th>$j$</th>
<th>$s_j^0$ [days]</th>
<th>$s_j^1$ [days]</th>
<th>$\Delta s_j$ [days]</th>
<th>$\sigma_j$ [days]</th>
<th>$w_j$</th>
<th>$\text{float}_j$ [days]</th>
<th>$\delta_j$ [days]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Proposed method without improvement procedure</td>
<td>Proposed method with improvement procedure</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>6.4192</td>
<td>0.4192</td>
<td>0.5714</td>
<td>14.9351</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>11</td>
<td>11.8004</td>
<td>0.8004</td>
<td>0.8146</td>
<td>0</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>22</td>
<td>23.1165</td>
<td>1.1165</td>
<td>1.3333</td>
<td>25.5818</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>27</td>
<td>28.0487</td>
<td>1.0487</td>
<td>1.3391</td>
<td>0</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>32</td>
<td>32.9997</td>
<td>0.9997</td>
<td>1.3401</td>
<td>0</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>51</td>
<td>52.7565</td>
<td>1.7565</td>
<td>2.0191</td>
<td>0</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>59</td>
<td>61.1249</td>
<td>2.1249</td>
<td>2.0952</td>
<td>50.4624</td>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>6</td>
<td>6.4192</td>
<td>0.4192</td>
<td>0.5714</td>
<td>0</td>
<td>39</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>32</td>
<td>32.9997</td>
<td>0.9997</td>
<td>1.3401</td>
<td>25.1000</td>
<td>16</td>
<td>3</td>
</tr>
<tr>
<td>12</td>
<td>51</td>
<td>52.7565</td>
<td>1.7565</td>
<td>2.0191</td>
<td>0</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>56</td>
<td>58.2034</td>
<td>2.2034</td>
<td>2.1632</td>
<td>0</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>68</td>
<td>70.8522</td>
<td>2.8522</td>
<td>2.3372</td>
<td>0</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>34</td>
<td>35.3795</td>
<td>1.3795</td>
<td>1.4591</td>
<td>51.8100</td>
<td>16</td>
<td>5</td>
</tr>
<tr>
<td>16</td>
<td>10</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>2</td>
</tr>
</tbody>
</table>

Buffer sizes were found by solving the model given by equations (4)–(10) and (11)–(12) by means of Lp_solve software. Then the procedure of improving the solution was applied. The results of calculations are presented in Table 2 (first variant) and Table 3 (second variant). A bar chart of the buffered schedules is presented in Figs 2 and 3.

Table 3. Results of calculations by the proposed method – the second variant

<table>
<thead>
<tr>
<th>j</th>
<th>( s_0 ) [days]</th>
<th>( s_1 ) [days]</th>
<th>( \Delta s_j ) [days]</th>
<th>( \sigma_j ) [days]</th>
<th>( w_j )</th>
<th>( \text{float}_j ) [days]</th>
<th>( \delta_j ) [days]</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>56</td>
<td>58.2034</td>
<td>2.2034</td>
<td>2.1632</td>
<td>0</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td>47</td>
<td>48.9383</td>
<td>1.9383</td>
<td>1.9910</td>
<td>0</td>
<td>16</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>75</td>
<td>77.5163</td>
<td>2.5163</td>
<td>2.5039</td>
<td>100.2802</td>
<td>5</td>
<td>3</td>
</tr>
</tbody>
</table>
Buffer sizes calculated using the *adapted float factor heuristic* are as follows: $\delta_3 = 1$, $\delta_5 = 3$, $\delta_9 = 4$, $\delta_{11} = 7$, $\delta_{15} = 5$, $\delta_{18} = 3$, $\delta_j = 0 \ \forall j \in V \setminus H$. Using the *starting time criticality heuristic*, following buffer sizes were obtained: $\delta_3 = 1$, $\delta_5 = 1$, $\delta_9 = 3$, $\delta_{11} = 3$, $\delta_{15} = 2$, $\delta_{18} = 3$, $\delta_j = 0 \ \forall j \in V \setminus H$.

Table 4 lists values of the instability cost function, established by simulation, of schedules with buffer sizes calculated by means of the proposed method, the *adapted float factor*, and *starting time criticality heuristics*. Application of all these procedures significantly improves the stability of the schedule – the instability cost is over four times lower than in the unbuffered schedule. The solution obtained by means of the proposed method with improvement procedure is better than solutions obtained by the *starting time criticality heuristic* (most widely presented in the literature).
5. Conclusions

A stable baseline schedule is crucial for the success of construction projects. Changes to the program introduced in reaction to process duration disruptions result in difficulties with managing subcontractors and suppliers. Furthermore, not keeping to the contract due date and milestones is usually connected with penalties.

Instability of the schedule may seriously affect continuity of works, labor and delivery cost, management effort, etc. Proper buffer sizing and placement enables the planner to protect the project against disruptions that occur during execution and to minimize the cost of not keeping to the baseline. The scale of the schedule instability cost is affected also by the policy of starting the processes: the railway policy is related with greater cost, while the possibility to start activities as soon as their predecessors finish reduces the possibility of delaying the activities to follow.

Fig. 3. Project schedule with buffers sizes obtained using the proposed method and the improvement procedure – the second variant; gray bars represent buffers in front of processes
The proposed method seems better than simple heuristics that do not allow for the probabilistic nature of process duration and are applicable only in the case of the railway policy. The float distribution rule employed by the proposed approach considers both duration variability and propagation of disruptions in the network. As the example illustrates, the method provides results similar in their quality to the results of the best known heuristic procedures. Its considerable advantage is the possibility to analyze resource constraints and to improve resource allocation (i.e. to change the resource allocation adopted in the baseline schedule). In contrast, methods presented in the literature on the subject consider only buffer allocation in baseline schedules and provide resource constraints feasible solutions while maintaining the initial resource flow. The proposed improvement procedure has the potential of providing solutions superior to those obtained by the existing heuristics.

Table 4. Comparison of schedule (solution) robustness obtained using different methods

<table>
<thead>
<tr>
<th>Method used to buffer sizing</th>
<th>Instability cost</th>
<th>Instability cost</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Railway policy for all activities starts (first variant)</td>
<td>Railway policy only for activities $j \in H$ (second variant)</td>
</tr>
<tr>
<td>Proposed method with improvement procedure</td>
<td>10.42</td>
<td>8.33</td>
</tr>
<tr>
<td>Proposed method without improvement procedure</td>
<td>10.65</td>
<td>9.84</td>
</tr>
<tr>
<td>Adapted float factor heuristic</td>
<td>12.57</td>
<td>11.47</td>
</tr>
<tr>
<td>Starting time criticality heuristic</td>
<td>10.52</td>
<td>8.42</td>
</tr>
<tr>
<td>Unbuffered schedule</td>
<td>41.50</td>
<td>38.11</td>
</tr>
</tbody>
</table>

Acknowledgements

The work was financed by the Polish Ministry of Science and Higher Education (grant no. N N506 254637).

References


---

STATYBOS VYKDYMO GRAFIKO STABILUMO UŽTIKRINIMAS PASKIRSTANT LAIKO REZERVUS

P. Jaškowski, S. Biruk

Santrauka. Tikrasis statybos projektų baigimo laikas retai atitinka pradinius užsibrėžtus planus. Taip yra dėl nenumatytytų atvejų, kurie daro įtaką galutiniams projektų laikui. Šiame straipsnyje autorai pristato metodą, kuris pagerina stabilitę statybos grafikų, padedančius sumažinti neigiamus atsitiktinių trikdžių poveikį vykdant projektą. Autoriai pritaiko

Reikšminiai žodžiai: statybos vykdymo grafikas, tvarus sprendimas, patikimas grafikas, rezervas, rizikos valdymas, projektų valdymas, tiesinis programavimas.

Piotr JAŚKOWSKI. Ph. Dr at Lublin University of Technology, Poland. He is a Head of the Group of Construction and Investments Processes Engineering working in the Building Institute. He is a member of the Section of the Engineering of Construction Projects in Civil Engineering Committee of Polish Academy of Science. He is the member of EURO Working Group “OR in Sustainable Development and Civil Engineering” and International Group on Reliability. His research interests include the organization and modeling of construction processes, construction projects scheduling, and optimization techniques.

Sławomir BIRUK. Ph. Dr at Lublin University of Technology, Poland. He is the member of EURO Working Group “OR in Sustainable Development and Civil Engineering” and International Group on Reliability. His research interests include the organization and modeling of construction processes, construction projects scheduling, optimization techniques, simulation modeling, construction methods and equipment.