BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 65, No. 6, 2017
DOI: 10.1515/bpasts-2017-0097

Financial optimisation of the scheduling for the multi-stage project

M. KLIMEK[1]* and P. ŁEBKOWSKI[2]

[1]State School of Higher Education of Biala Podlaska, Poland
[2]AGH University of Technology of Cracow, Poland

Abstract. The paper analyses the problem of discounted cash flow maximising for the resource-constrained project scheduling from the project contractor's perspective. Financial optimisation for the multi-stage project is considered. Cash outflows are the contactor's expenses related to activity execution. Cash inflows are the client's payments for the completed milestones. To solve the problem, the procedure of backward scheduling taking into account contractual milestones is proposed. The effectiveness of this procedure, as used to generate solutions for the simulated annealing algorithm, is verified with use of standard test instances with additionally defined cash flows and contractual milestones.

Key words: resource-constrained project scheduling problem, discounted cash flows, milestones, backward scheduling, simulated annealing.

1. Introduction

The resource-constrained project scheduling problem (RCPSP) has been studied for a number of years, given its practical applicability. Numerous optimisation models have been analysed, with various resource categories and various activity execution modes [1–3]. The optimisation criteria used most often are those time-related, e.g. minimising makespan, or financial criteria, such as discounted cash flow maximising (RCPSP with discounted cash flows – RCPSP-DC).

In practice, a material aspect of project planning is an analysis of financial effects of the decisions made. For this reason, numerous research papers include project-related cash flows in project scheduling. For long-term projects, the cash flows are discounted; their net present value (NPV) is computed at the assumed discount rate. Russell was the first to propose, in 1970, that NPV maximising should be included in the project scheduling problem; he suggested the max-NPV model [4]. First research papers covering NPV optimisation did not take into consideration project resource constraints [4–5]. Currently, the research focuses on the RCPSP-DC problem [6–20]. For a review of models and algorithms including cash flow optimisation, we refer the reader to [1–3, 21]. In this paper, selected RCPSP-DC problems are discussed only, those material to the model analysed herein.

While computing NPV for a project, all related cash flows are taken into consideration; they are discounted to reflect changes in time value of money. Subject to optimisation is, inter alia, the schedule of payments for completed project activities or milestones. As part of payment project scheduling (PPS), various models are analysed of settlements between the contractor and the client [12, 14, 22–25]. The objective is to find a payment schedule which would maximise NPV from the contractor's and/or client's perspective. The elements to be determined include the aggregate amount of the client's payments for project execution, number of payment tranches, amounts of individual tranches, tranche payment deadlines, etc. Amounts and deadlines of individual tranches may take into consideration various factors, including activity execution cost or progress of project implementation. The client's and the contractor's interests diverge: the contractor would like the client's payments to be as early as possible (this translates into larger payments in terms of their NPV), while the client would rather defer payments. The PPS problem is analysed from the contractor's perspective [14, 25] or the client's [23–24]. Solutions which would be satisfactory to the both client and contractor are also sought for [22, 26].

This paper considers the PPS problem from the perspective of the contractor. In such approach, the research into the RCPSP-DC problem covers cash inflows and cash outflows. Cash outflows are the contractor's expenses most often relating to activity execution and resource use. Cash inflows are, as a rule, the client's payments to the contractor for the executed project, its milestones or activities. Various payment models have been considered [12, 14, 25], including:

- lump-sum payment (LSP) model – with the client's one-off payment after the completion of the entire project;
- payments at event occurrences (PEO) model – payments are triggered by events: completion of a milestone or an activity; in the latter case we have to do with payments at activities' completion times (PAC) model;
- equal time intervals (ETI) model – payments are made at fixed time intervals, with the number of payments predefined;
- progress payments (PP) model – payments are made at fixed time intervals, with no fixed number of payments.

For the purposes of project settlement between the client and the contractor, a bonus-penalty model is also analysed with predefined bonuses and penalties [7–11, 27-29]. Penalties

*e-mail: m.klimek@dydaktyka.pswbp.pl

Manuscript submitted 2016-12-06, revised 2017-04-05, initially accepted for publication 2017-05-22, published in December 2017.

are imposed for delayed completion of a project or milestone, while bonuses are granted for early completion of a project or milestone. There are defined exact execution deadlines or time windows in which the execution of an activity or milestone is neither rewarded, nor penalised. A problem is analysed [7], where for each activity a deadline for its execution is defined in order to minimise the NPV of expense on activity execution before or after the required deadline (earliness-tardiness penalties). Various contractor-client settlement systems with progress-of-execution-related penalties and bonuses are considered [8–11, 27–28]. A penalty and bonus system is designed to motivate the contractor to execute the work as soon as possible. The absence of such supplementary incentives for contractors may lead to delays in project execution. For a penalty and bonus system to be efficient from the contractor's perspective, benefits from bonuses should exceed the contractor's additional expenses on early execution of activities (milestones), while tardiness penalties should exceed the contractor's savings generated by delayed execution of activities (milestones). For a penalty and bonus system to be efficient from the client's perspective, bonus paid to the contractor for early execution of activities (milestones) should not exceed the client's profit from the early execution, while penalty for a delayed execution of work should exceed the benefits lost by the client due to a delay. The differences between the contractor's and the client's preferences pertaining to, for instance, execution deadlines and amounts of individual payments for project execution may decide whether a penalty and bonus system is practically applicable to a given project.

In this paper, the authors propose project settlement by milestones. In such system, the client and contractor agree upon project milestones and define deadlines and payments for the timely execution thereof. For any delay, a penalty is imposed on the contractor, reducing the client's payment for the delayed milestone. The contractor's benefits from milestone-based settlement include earlier payments for the work executed, and amounts of such payments can be applied towards financing Just-In-Time operations (execution of further activities, purchase of materials etc.).

On the other hand, not all researchers/practitioners recommend the use of milestones. According to Goldratt, and the critical chain project management (CCPM) concept [29–30], it is advisable to eliminate milestones, because their use may lead to delays in project implementation. In our work during the stage execution, activities from the next stage can be performed, thus avoiding waste of time when running a project with defined milestones.

A system of project settlement by predefined milestones may have practical applications. In project execution, milestone-based approach is used in practice in the progress of execution measured by predefined stages, with milestones defined as stage completions. Milestones are coordination and monitoring points of special importance, given the weight of the completion of project stages on further stages and activities, as well as on the final outcome (final acceptance). Milestone defining is a material element of project planning. The number of milestones depends on the specific nature of the project, but it should not be too large, because a milestone should remain an outstanding and exceptional event in project execution. Project scheduling is supported by IT systems (such as Microsoft Project), which often provide for using the milestone technique, thus facilitating project management and enhancing control of timely project execution; it can also support the contractor-client settlement.

The discounted cash flow optimising for the resource constrained project scheduling problem settled by milestones, in the model considered herein, has as yet been researched, by the authors only. Other authors have analysed, e. g., multi-mode RCPSP (MMRCPSP) [27–28] or discrete time/resource trade-offs problem (DTRTP) [31] with settlement by milestones.

In this paper, the RCPSP problem with predefined contractual milestones and a single activity execution mode (single-mode RCPSP), and with financial settlements by milestones [8–11], where the objective function is maximising aggregate discounted cash flows: with the client's payments for completed project stages (milestones), system of penalties for delayed achievement of milestones and the contractor's outflows related to the activities performed.

The purpose of this paper is to present a new backward scheduling procedure with optimisation of completion times of contractual milestones, as well as to prove that the procedure is efficient for the analysed model of project with settlements by milestones. A numerical example is also presented to illustrate the problem and the procedure. The paper concludes with the presentation of results of numerical experiments for test problems from project scheduling problem library (PSPLIB) [32], with additionally defined cash flows and contractual settlement milestones.

2. Formulation of the problem

The nonpreemptive single-mode RCPSP problem is considered here. The project is modelled as an acyclic directed graph $G(V, E)$ in the AoN (acivity-on-node) representation; V stands for the set of nodes (vertices) and E stands for the set of arcs describing precedence relations between activities.

The authors' model is analysed of NPV maximising with work settlement by milestones: with cash flows defined for activities and contractual milestones [8–11]. The model may be described as follows (1–5):

Maximise

$$F = \sum_{i=1}^{N_A} \left(CFA_i \cdot e^{-\alpha \cdot ST_i}\right) + \sum_{m=1}^{N_M} \left(CFM_m \cdot e^{-\alpha \cdot MT_m}\right), \tag{1}$$

at the following constraints:

$$\forall (i, j) \in E : ST_i + d_i \leq ST_j, \tag{2}$$

$$\forall t \, \forall k : \sum_{i \in J(t)} r_{ik} \leq R_k, \tag{3}$$

and for predefined contractual milestones:

$$MT_m = \max_{i \in MA_m} (FT_i), \tag{4}$$

$$CFM_m = MP_m - MC_m \cdot \max(MT_m - MD_m, 0), \tag{5}$$

where:

- N_A stands for the number of project activities,
- i stands for the index of a project activity, $i = 1, \ldots, N_A$,
- N_M stands for the number of project contractual milestones,
- m stands for the index of a project milestone, $m = 1, \ldots, N_M$,
- d_i stands for the duration of activity i,
- α stands for the discount rate in the same period of capitalisation as activity starting times, milestones completion times etc. are expressed (months, days),
- ST_i stands for the planned starting time of activity i,
- FT_i stands for the planned finish time of activity i; ($FT_i = ST_i + d_i$),
- k stands for the index of a resource type, $k = 1, \ldots, K$ (K is the number of resource types), each resource is renewable and it has a maximum capacity and at any moment no more than this amount can be in use,
- R_k stands for the number of type k resources available at any time during project execution,
- r_{ik} stands for the number of type k resources used to execute activity i,
- $J(t)$ stands for the set of activities performed in the time interval $[t - 1, t]$,
- CFA_i stands for the contractor's expenses incurred for the execution of activity i at its start,
- CFM_m stands for the client's payment for the execution of milestone m as determined in the current schedule (the contractor's inflows),
- MD_m stands for the contractual deadline for milestone m,
- MT_m stands for the planned (in the current schedule) finish time of milestone m,
- MA_m stands for the set of activities performed in milestone m, with the defined contractual deadline for those activities equal to MD_m,
- MP_m stands for the client's contractual payment for the execution of milestone m,
- MC_m stands for the contractual unit cost of delay for the delayed execution of milestone m.

A solution is defined as a vector whose components ST_i are activity starting times and for which the objective function F (1) takes the maximum value, taking into consideration:

- finish-start zero-lag precedence relations occurring between activities (2),
- limited quantities of renewable resources (humans, machines etc.) used to perform activities (3),
- financial settlements related to project contractual milestones (4–5).

In financial settlements of project work, from the contractor's perspective, the contractor's cash inflows are the client's payments CFM_m (for milestones $m = 1, \ldots, N_M$), while cash outflows are expenses incurred in connection with the performance of activities CFA_i (for activities $i = 1, \ldots, N_A$).

In the calculation of the client's payments CFM_m (4), there are taken into consideration amounts MP_m payable for the execution of project milestones, reduced if a given contractual deadline MD_m is not met (default penalties are determined based on the contractual unit cost of delay MC_m). No bonus (increased payment to the contractor) is provided for an early execution of a milestone. In the event of an early execution, the contractor's benefit is an earlier acquisition of funds, thus of a of higher discounted value. It is assumed that the client makes payments CFM_m exactly at the time MT_m provided in the current schedule for the execution of a milestone (the problem of the client's delayed payments is not included in the model examined, as it is beyond the contractor's control; it may be assumed that the client is obliged to compensate any delayed payment with additional payment).

For simplicity the model only taking into account contractor's costs that can be qualified as the cash outflows. All of the contractor's outflows are related to the expenses on the execution of a given activity (CFA_i), connected with using resources and materials, transportation thereof etc. Such outflows occur at the time of a planned start of a given activity in accordance with the current schedule.

The proposed model of project settlement by milestones is favourable to the contractor, who in this way receives, from the client, funds for financing the execution of further activities, purchase of materials etc. before the final completion of the project. The client makes payments before the final completion to the client's disadvantage. On the other hand, project settlement by milestones gives the client control of the progress of work during the project execution, while the contractual penalty system urges the contractor to execute project milestones in a timely manner.

With a view to explaining the problem proposed, we present the following example. Let us assume that a project comprises eight activities performed with a single resource type, whose availability is 10. The AoN activity network for the project with the aforementioned parameters fed into the optimisation model examined is presented in Fig. 1. Activities 0 and 10 are dummy

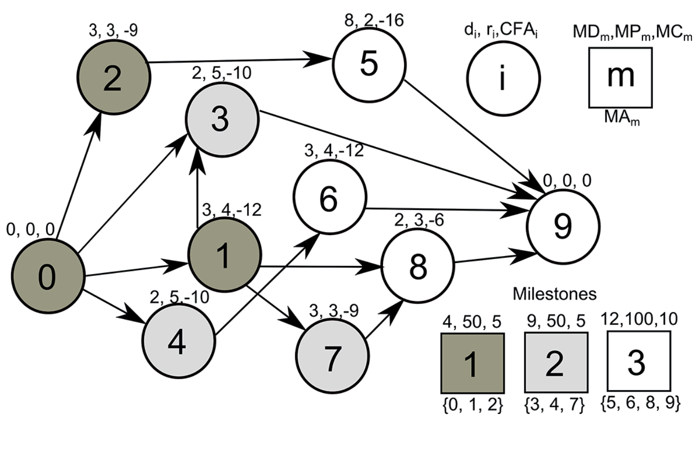

Fig. 1. Activity network for a project with contractual milestones

activities representing the initial and final, respectively, vertices of the graph $G(V, E)$.

The following three contractual stages (concluding with milestones) are defined for the project:

- Milestone 1, by which activities $MA_1 = \{0, 1, 2\}$ are to have been executed, with the contractual deadline $MD_1 = 4$; the client's payment for the milestone is $MP_1 = 50$; the payment may be reduced by the cost of delay, if any, calculated based on the unit cost $MC_1 = 5$;
- Milestone 2, by which activities $MA_2 = \{3, 4, 7\}$ are to have been executed, with the contractual deadline $MD_2 = 9$; the client's payment for the milestone is $MP_2 = 50$; the payment may be reduced by the cost of delay, if any, calculated based on the unit cost $MC_2 = 5$;
- Milestone 3, by which activities $MA_3 = \{5, 6, 8, 9\}$ are to have been executed, with the contractual deadline $MD_3 = 12$; the client's payment for the milestone is $MP_3 = 100$; the payment may be reduced by the cost of delay, if any, calculated based on the unit cost $MC_3 = 10$.

To compute the discounted value of cash flows, the discount rate of $\alpha = 0.01$ has been used, with the accretion period of 1.

In order to explain the way of determining the objective function F (1), we will use an example of a schedule which minimises the project makespan (Fig. 2).

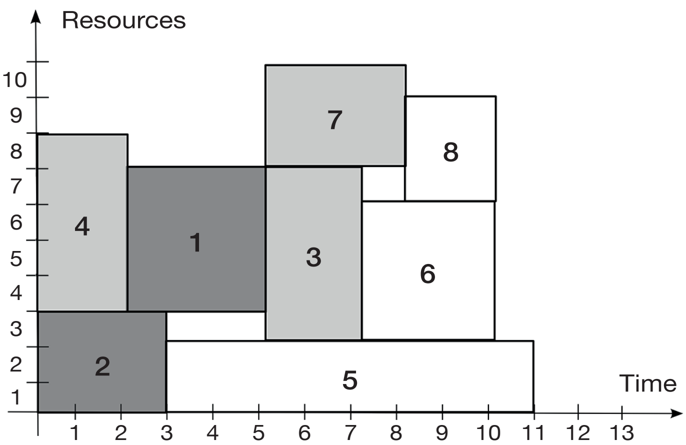

Fig. 2. Example of a schedule for a project analysed

The schedule in Fig. 2 provides for a delayed execution of Milestone 1 (actual finish time $MT_1 = 5$ against contractual deadline $MD_1 = 4$), which reduces the client's payment for the milestone by 5 monetary units. The other milestones are completed before their contractual deadlines as it increases the discounted value of the client's payments. The contractor's discounted inflows for executed milestones are:

$$\sum_{m=1}^{N_M}\left(CFM_m \cdot e^{-\alpha \cdot MT_m}\right) = \left(50 - 5\right) \cdot e^{-0.05} + 50 \cdot e^{-0.08} + \\ + 100 \cdot e^{-0.11} = 178.54.$$

The contractor's discounted outflows related to activity execution are:

$$\sum_{i=1}^{N_A}\left(CFA_i \cdot e^{-\alpha \cdot ST_i}\right) = -12 \cdot e^{-0.02} - 9 \cdot e^{0} - 10 \cdot e^{-0.05} - 10 \cdot e^{0} + \\ + -16 \cdot e^{-0.03} - 12 \cdot e^{0.07} - 9 \cdot e^{-0.05} - 6 \cdot e^{-0.08} = -81.09.$$

Summing up, the objective function is as follows:

$$F = \sum_{i=1}^{N_A}\left(CFA_i \cdot e^{-\alpha \cdot ST_i}\right) + \sum_{m=1}^{N_M}\left(CFM_m \cdot e^{-\alpha \cdot MT_m}\right) = \\ = -81.09 + 178.54 = 97.45.$$

In the model under analysis, it is favourable for the contractor to schedule its expenses (related to activity execution and incurred at the launch of the activity) as late as possible and to acquire the client's payments (made upon project milestone completion) as soon as possible. The advisable strategy is the "as late as possible" (ALAP) strategy, taking into consideration project milestones, which should be completed as soon as possible. To the best of the authors' knowledge, no procedure is known (other than those developed by the authors [8–11] which would generate activity arrangements appropriate for the optimisation model analysed.

3. Methods for solving the problem

In local search algorithms for the RCPSP problem, indirect representations of the problem are used, such the activity list, used herein, where a solution is a sequence of numbers of consecutive activities, arranged to reflect order relations. A solution in the form of an activity list is decoded into a workable schedule in the direct representation (of activity starting times) with use of the schedule generation schemes (SGS) [33]. For the deterministic RCPSP problem, the decoding procedures used include:

- serial SGS – a procedure in which, for each consecutive time *t*, the starting time is determined for the first not yet arranged activity in the activity list, so that the time is as soon as possible, given the precedence and resource constraints;
- parallel SGS – a procedure in which, at each consecutive time *t*, these activities (analysed in the order defined in the activity list) are started which have not yet been arranged and may be started at a given *t*, given the precedence and resource constraints.

SGS procedures usually support forward scheduling. However, backward scheduling may also be used, in which decoding procedures determine activity starting times from the bottom of the activity list up, commencing scheduling from the contractual final deadline of the project.

For the problem of maximising net discounted cash flows, procedures are used combining techniques of generating both backward and forward scheduling (bidirectional schedule generation scheme [15]): the activities to which net aggregate inflows are ascribed are scheduled as soon as possible, while the activities to which net aggregate outflows are ascribed are scheduled as late as possible.

Besides transformed decoding procedures, discounted cash flow optimisation is also performed with use of improvement procedures, providing for right shifts of activities with net aggregate outflows (for solutions generated with forward scheduling) and/or left shifts of activities with net aggregate inflows (for solutions generated with backward scheduling). Research covers various solution improvement procedures (reviewed in [18]) using activity shifts. However, those procedures are not compatible with the model of activity settlement by milestones, discussed here (see Formulae 1–5). They are used to RCPSP-DC models in which cash outflows and/or inflows are ascribed to activities [18–20, 25, 34–36].

In optimisation model with the objective function F cash inflows (the client's payments for executed milestones of the project, made exactly at the completion times) should be arranged as soon as possible, while cash outflows (the contractor's expenses incurred at the starting times of activities) as late as possible. Any increase in the value of the objective function is achieved by postponing those activities (as cash outflows are ascribed to them) whose delayed start does not change the milestone completion time. It might be favourable to postpone a milestone completion time if it increases benefits from delayed expenditure incurred at the start of activity execution. For the problem considered, the issues analysed by the authors include:

- use of various procedures of forward and backward schedule decoding, taking into consideration project contractual milestones;
- use of various improvement algorithms, in which activities are right shifted in forward scheduling and left shifted in backward scheduling; activity shifts are performed against precedence and resource constraints for solutions with predefined resource allocation, or with use of a modified SGS decoding procedure.

The existing research results indicate that best results are obtained for a backward scheduling strategy including the selection of most favourable completion times for project contractual milestones [8]. In this paper, a new backward scheduling procedure is proposed, dedicated to the optimisation model analysed.

A technique called forward-backward improvement (FBI) [37–40] or justification methods [41–45] can also be useful in improving solution.

The FBI procedure iteratively applies forward and backward scheduling until no further improvement in the quality of project (e.g. makespan or NPV of a project) is feasible.

Given a schedule, the left (right) justification of an activity involves determining the start time for a given activity as early (late) as possible, taking into account the resource and precedence constraints, while the start times of the other activities remain unchanged. The left (right) justification by extremes of a schedule is achieved by the left (right) justification of each activity, in the increasing (decreasing) order of the start (end) times of activities. Activities justification can be performed in a different order, the justification by eligibles: an activity is eligible for left (right) justification if its predecessors (successors) and any activity that starts before (ends after) any predecessors (successors) are left (right) justified [42].

Justification is a simple technique that can be easily incorporated in various algorithms and models for RCPSP improving the quality of the schedules generated [39]. Algorithms applicable to the problem analysed include:

- forward scheduling with the right justification by extremes or by eligible,
- backward scheduling with double (left and then right) justification by extremes or by eligible.

Right activity justification should take into account contractual deadlines of a project. Implementing algorithms with justification techniques will be the subject of the authors' further work.

4. Backward scheduling procedure

The proposed procedure uses backward scheduling, taking into consideration project contractual milestones. A solution is written as a list of activities, decoded using an appropriately transformed serial or parallel SGS, taking into consideration predefined milestone deadlines. Activities are scheduled from the bottom of the list of activities up, starting from the deadline for the completion of the last project milestone. For each activity, such starting time for its execution is selected which, given the precedence and resource constraints, supports the completion of project milestones by the predefined times $MT_m (m = 1, \ldots, N_M)$. If for a given list of activities and predefined times MT_m the generated schedule is incorrect (e.g., one of the activities is ascribed a negative starting time), then the schedule is corrected by right shifting all activities, until a correct arrangement appears. The proposed backward scheduling procedure with optimisation of completion times of project contractual milestones is presented in Fig. 3. In consecutive runs of the procedure, unit left shifts of project milestones are performed, until the largest value of the objective function F is obtained.

```
FOR m := 1 TO Nm DO
   MT[m] := MD[m];
S* := SGS(x, MT);
FOR m := 1 TO Nm DO
BEGIN
   MT[m] := MT[m] - 1;
   S := SGS(x, MT);
   IF(F(S) > F(S*)) THEN
   BEGIN //increase in F
         S* := S;
         m := m - 1;
         //further improvement at a given
         milestone
   END
   ELSE //no increase in F
         MT[m] := MT[m] + 1;
         //undo unit shift at the milestone m
END;
Return schedule S*
```

Where: x stands for the currently analysed activity list, MT stands for the table of predefined MT_m for project milestones $(m = 1, \ldots, N_M)$, $F(S)$ stands for the value of objective function F at the schedule S, $SGS(x, MT)$ is the backward scheduling procedure taking into consideration the predefined MT_m for project milestones for the solution (activity list) x, S^* stands for the best currently identified schedule with the highest value of the objective function F, S stands for the currently analysed schedule generated with the $SGS(x, MT)$ procedure.

Fig. 3. Backward scheduling procedure with the optimisation of completion times of contractual milestones

The procedure presented in Fig. 3 is an improvement of the algorithm of backward scheduling with the project milestone deadlines generated for a forward schedule maximising the client's total discounted payments CFM_m, with expenditure on activity execution not included.

The procedure uses the backward scheduling technique and (serial or parallel) SGS schemes, which generate a schedule by analysing consecutive activities backwards, from the bottom of the activity list analysed, taking into consideration the predefined MT_m. An example of a schedule generated for the activity list $x = \{1, 4, 2, 5, 6, 3, 8, 7\}$ and predefined project milestones completion time $MT_1 = MD_1 = 4$, $MT_2 = MD_2 = 9$, $MT_3 = MD_3 = 12$ is presented in Fig. 4a. The arrangement improved with use of left shifts of project milestones with new milestones completion time $MT_1 = 3$, $MT_2 = 7$, $MT_3 = 11$ is presented in Fig. 4b.

The schedule in Fig. 4a was backward generated. Each activity is started as late as possible, taking into consideration the contractual project milestone deadlines MD_m. The client's discounted payments for executed milestones are:

$$\sum_{m=1}^{N_M}\left(CFM_m \cdot e^{-\alpha \cdot MT_m}\right) = 50 \cdot e^{-0.04} + 50 \cdot e^{-0.09} + 100 \cdot e^{-0.12} = \\ = 182.43.$$

They are higher than for the schedule in Fig. 2, because all project milestones have been completed by their respective contractual deadlines. While for the schedule in Fig. 2, compared with that in Fig. 4a, the contractor benefits from earlier payments for Milestones 2 and 3 (their discounted values are higher), the benefit is more than offset by contractual penalty for late execution of Milestone 1.

The contractor's discounted expenditure on activity execution is:

$$\sum_{i=1}^{N_A}\left(CFA_i \cdot e^{-\alpha \cdot ST_i}\right) = -12 \cdot e^{-0.01} - 9 \cdot e^{0.01} - 10 \cdot e^{-0.07} + \\ + -10 \cdot e^{0.05} - 16 \cdot e^{-0.04} - 12 \cdot e^{0.09} - 9 \cdot e^{-0.06} - 6 \cdot e^{-0.1} = -79.87.$$

The expenditure is lower than for the schedule in Fig. 2, owing to later starts of some activities.

The value of objective function is:

$$F = \sum_{i=1}^{N_A}\left(CFA_i \cdot e^{-\alpha \cdot ST_i}\right) + \sum_{m=1}^{N_M}\left(CFM_m \cdot e^{-\alpha \cdot MT_m}\right) = \\ = -79.87 + 182.43 = 102.56.$$

An improvement procedure for a backward generated solution consists in a 'profitability' analysis of unit time shifts of the planned milestone deadlines MT_m, starting from the first one and ending at the last milestone. For the activity list $x = \{1, 4, 2, 5, 6, 3, 8, 7\}$, the run of the proposed lest shifts of milestones is as follows:

- **Left shift of Milestone 1**. The left shift of the milestone completion time by a unit (so that $MT_1 = 3$) increases the value of the objective function F from 102.56 to 102.83, owing to an increased discounted value of the payment for Milestone 1. It is not feasible to further left-shift Milestone 1 completion time to $MT_1 = 2$, given the duration of activities executed by this milestone ($d_1 = 3$, $d_2 = 3$); the improvement algorithm moves to the analysis of Milestone 2.
- **Left shift of Milestone 2**. The left shift of the milestone completion time by a unit (so that $MT_2 = 8$) increases the value of the objective function F from 102.83 to 103.01, owing to an increased discounted value of the payment for Milestone 2; the left shift of the milestone completion time by two units (so that $MT_2 = 7$) again increases the value of the objective function F, now from 103.01 to 103.20. It is not feasible to further left-shift Milestone 2 completion time to $MT_2 = 6$, because then the starting time for Activity 1 or Activity 2, determined by backward procedure, would be negative; the improvement algorithm moves to the analysis of Milestone 3.
- **Left shift of Milestone 3**. The left shift of the milestone completion time by a unit (so that $MT_3 = 11$) increases the value of the objective function F from 103.20 to 103.77, owing to an increased discounted value of the payment for

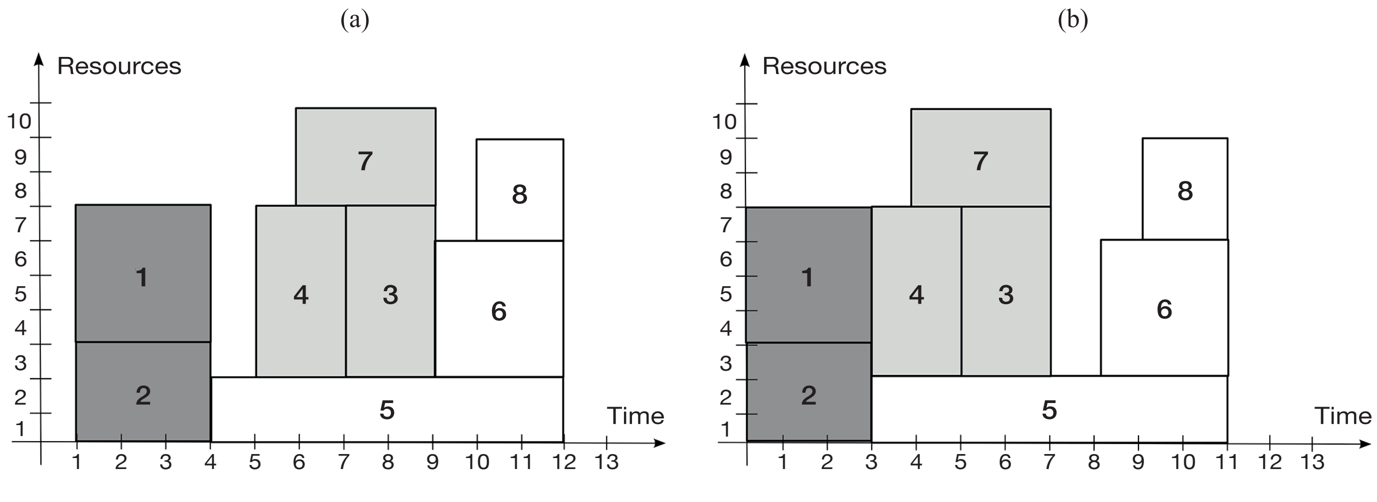

Fig. 4. Schedules generated by serial SGS: 4a) assuming $MT_1 = MD_1 = 4$, $MT_2 = MD_2 = 9$, $MT_3 = MD_3 = 12$; 4b) assuming earlier completion of certain milestones $MT_1 = 3$, $MT_2 = 7$, $MT_3 = 11$

Milestone 3. It is not feasible to further left-shift Milestone 3 to $MT_3 = 10$, because then the starting time for Activity 1 or Activity 2, determined by backward procedure, would be negative; the improvement algorithm stops.

Left shifts of project milestones having been performed, the schedule presented in Fig. 4b is generated. For this schedule, the contractor's discounted expenditure on activity execution is:

$$\sum_{i=1}^{N_A}\left(CFA_i \cdot e^{-\alpha \cdot ST_i}\right) = -12\cdot e^{0} - 9\cdot e^{0} - 10\cdot e^{-0.05} - 10\cdot e^{-0.05} + \\ + -16\cdot e^{-0.03} - 12\cdot e^{0.08} - 9\cdot e^{-0.04} - 6\cdot e^{-0.09} = -80.95.$$

The contractor's discounted expenditure is here higher than for the schedule in Fig. 4a, due to earlier start of project milestones, which have been left shifted. On the other hand, these shifts increase the client's discounted payments for executed milestones:

$$\sum_{m=1}^{N_M}\left(CFM_m \cdot e^{-\alpha \cdot MT_m}\right) = 50\cdot e^{-0.03} + 50\cdot e^{-0.07} + 100\cdot e^{-0.11} = \\ = 184.72.$$

The aggregate value of the objective function is higher for the schedule in Fig. 4b than the schedule in Fig. 4a:

$$F = \sum_{i=1}^{N_A}\left(CFA_i \cdot e^{-\alpha \cdot ST_i}\right) + \sum_{m=1}^{N_M}\left(CFM_m \cdot e^{-\alpha \cdot MT_m}\right) = \\ = -80.95 + 184.72 = 103.77.$$

Computations show that for the illustrative project, scheduling in line with the ALAP principle, with contracted milestone deadlines, has no advantage. For the schedule in Fig. 4a the value of the objective function $F = 102.56$ is by 1.19 lower than for the schedule in Fig. 4b ($F = 103.77$). Using a procedure with the optimisation of project milestones completion time increases the project's NPV, owing to the client's larger discounted payments for project milestones executed earlier than originally agreed, despite the contractor's larger discounted expenditure on early execution of activities.

5. Simulated annealing algorithm

The RCPSP is NP-hard [46]. Therefore, it is reasonable to generate a solution with heuristics. Review papers [3,47–48] contain an effectiveness analysis for algorithms used to solve the RCPSP problem. In this paper, the authors use a type of metaheuristics: simulated annealing (SA) algorithm, which is effective and has repeatedly been used to solve the RCPSP problem [47–49]. Solutions are represented as activity lists and decoded into a workable schedule with use of the (serial or parallel) SGS procedure which generates a backward schedule with the project milestone completion times optimisation, described in Section 4.

The SA algorithm developed by the authors (Fig. 5) is similar to the basic variant of this metaheuristics [50].

The procedure starts with initiating and tuning of SA Algorithm parameters, including the initial temperature T_0, final temperature T_k, current solution x etc. For the adopted initial solution x, a schedule S' is generated using backward sched-

```
Initiation;
Tuning;
S' := SGS(x);
T := T0;
REPEAT
BEGIN
    Select y a neighbour of x;
    S := SGS(y);
    IF(P(-F(S), -F(S'), T) > rand) THEN
    BEGIN
         x := y;
         S' := S;
    END;
    IF(F(S*) < F(S)) THEN
         S* := S;
    Update the current temperature T with the
    adopted cooling scheme;
END
UNTIL (T < Tk)
Return schedule S*
```

Where: T stands for the current temperature, T_0 stands for the initial (maximal) temperature, T_k stands for the final (minimal) temperature, *rand* is a random number in the interval (0, 1), x is the current solution in the form of an activity list, y is a neighbouring solution for x, $SGS(x)$ is a procedure for generating backward schedule with project milestones taken into consideration for the solution (activity list) x, $F(S)$ is the value of the objective function F at the schedule S, $P(-F(S), -F(S'), T)$ is the acceptance function for the solution y and the schedule S' generated for y, S is the currently analysed schedule, S' is the currently analysed schedule with time shifted start times for activity execution, S^* is the best current schedule.

Fig. 5. Simulated Annealing Algorithm

uling, taking into consideration project milestones, and the current temperature is set, equal to the initial temperature. Then, consecutive iterations of the algorithm are run, until the stop condition is met: the current temperature is equal to the final temperature T_k. Each iteration of the algorithm starts with the generation of a new schedule y in a neighbourhood of the current solution x; the rearrangement operators used are: swap, swap adjacent and insert. For the solution y, the backward decoding procedure is executed, taking into consideration project milestones and generating a schedule S. If the acceptance criterion $P(-F(S), -F(S'), T) = exp\left(\left(F(S) - F(S')\right)/T\right)$ is met, the solution y is written as the current solution x. If the schedule S generated for y is the best schedule since the algorithm start, it is memorised as S^*. Each iteration concludes with updating (reducing) the current temperature T; the cooling schemes used are: linear, logarithmic and geometric ones. At a given temperature T, a single solution is analysed, generated with backward procedure taking into consideration project milestones.

6. Numerical experiments

The experiments were run on a computer with a Intel Core I7, 3.0 GHz CPU and 8 GB RAM, using an application developed in C# in the Visual Studio.NET environment. For computations, 1,440 test instances from the PSPLIB [30] were used, from the sets J30 (30-activity projects), J60 (60-activity projects) and J90 (90-activity projects). For each project sourced from the PSPLIB, the authors assumed project settlement by three milestones [8]. For J30 projects, the set of activities executed in individual milestones were: $MA_1 = \{1, 2, \ldots, 10\}$, $MA_2 = \{11, 12, \ldots, 20\}$, $MA_3 = \{21, 22, \ldots, 30\}$. For J60 projects: $MA_1 = \{1, 2, \ldots, 20\}$, $MA_2 = \{21, 22, \ldots, 40\}$, $MA_3 = \{41, 42, \ldots, 60\}$ and for J90 projects: $MA_1 = \{1, 2, \ldots, 30\}$, $MA_2 = \{31, 32, \ldots, 60\}$, $MA_3 = \{61, 62, \ldots, 90\}$.

The contractual milestone deadlines were determined based on the project deadline *duedate* from a PSPLIB test instance:

$$MD_m = \left\lfloor \frac{1.4 \cdot duedate}{N_M} \cdot m \right\rfloor, \qquad m \in \langle 1, N_M \rangle. \tag{6}$$

For all test instances, the same parameters were used for financial settlements: $MP_1 = 40$, $MP_2 = 40$ and $MP_3 = 80$; $MC_1 = 1$, $MC_2 = 1$ and $MC_3 = 2$. Activity execution expenditure amounts CFA_i were computed *pro rata* to the aggregate demand for resources and the time necessary to execute a given activity, with the aggregate value for all activities set at 100:

$$CFA_i = \frac{d_i \cdot \sum_{k=1}^{K} r_{ik}}{\sum_{j=1}^{N_A} \left(d_j \cdot \sum_{k=1}^{K} r_{jk} \right)} \cdot 100, \quad \text{for} \quad i = 1 \ldots N_A. \tag{7}$$

For the experiments run, the discount rate $\alpha = 0.01$ was assumed.

The simulated annealing algorithm, with parameters chosen experimentally, was used to search the space of potential solutions with use of the procedures proposed. Various cooling schemes (logarithmic, geometric and linear), various moves (Swap, Swap Adjacent and Insert) and various initial temperatures were analysed. In the experiments described herein, solutions are generated with the SA configuration which has proved most effective for the problem studied: the geometric cooling scheme, Swap move and the initial temperature of 0.1 [8].

Table 1 presents the results of numerical experiments for 30-activity projects, Table 2 for 60-activity projects and Table 3 for 90-activity projects.

The numerical experiments were designed to verify the effectiveness of backward scheduling with the optimisation of contractual project milestones completion time (in the tables below: Alg2 with the serial SGS procedure and Alg4 with the parallel SGS procedure). For the sake of comparison, results are also presented for those algorithms of activity right shifts for a forward generated schedule with predefined resource allocation which proved most effective among the algorithms analysed by the authors [8–11] (Alg1 in Tables 1, 2 and 3). The experiments also covered backward scheduling, where a solution is backward generated with predefined (not optimised) project milestone deadlines set as the contractual deadlines (Alg3 with serial SGS and Alg5 with parallel SGS).

Given the stochastic nature of the SA algorithm, for each test instance and each procedure (Alg1–Alg5), three runs of the algorithm were executed, which adds up to a total of 1,440 runs of the SA algorithm for each procedure (Alg1–Alg5) of each of the sets J30, J60 and J90.

The average values of the objective function F for the best solutions found by all of the algorithms used are 21.03, 13.38 and 9.02 for the sets J30, J60 and J90, respectively. The use of procedures shifting milestone completion times increases the project NPV for each test instance.

Table 1
Results of numerical experiments for test projects from the set J30

Algorithm	t	F_{1000}	F_{5000}	F_M	F_A	$\#F_{best}$	$\%F_{best}$
Alg1	1.19	19.12	19.45	97.61	–78.16	97	–7.5%
Alg2	1.76	20.52	20.76	97.16	–76.40	1065	–1.3%
Alg3	0.06	15.97	16.23	86.50	–70.27	7	–22.8%
Alg4	8.55	14.33	14.41	88.18	–73.77	111	–31.5%
Alg5	0.26	9.29	9.29	78.90	–69.61	0	–55.8%

Here: t is the average time of algorithm run (in seconds), F_{1000} is the average value of the objective function F after 1,000 runs of the SA algorithm, F_{5000} is the average value of the objective function F after 5,000 runs of the SA algorithm, F_M is the average value of the objective function component based on the client's discounted payments CFM_m, F_A – based on the contractor's discounted expenditure CFA_i, $\#F_{best}$ is the number of solutions identical to the best solution found by all algorithms analysed (1,440 algorithm runs: 480 for test instances and three runs for each instance), $\%F_{best}$ is the mean deviation from the best solutions found by all of the algorithms analysed.

Table 2
Results of numerical experiments for test projects from the set J60

Algorithm	t	F_{1000}	F_{5000}	F_M	F_A	$\#F_{best}$	$\%F_{best}$
Alg1	7.23	8.76	9.21	80.70	–71.49	9	–31.1%
Alg2	6.25	12.26	12.91	80.92	–68.01	698	–3.5%
Alg3	0.16	7.08	7.72	68.45	–60.73	7	–42.3%
Alg4	60.96	5.90	6.27	70.85	–64.58	39	–53.1%
Alg5	0.36	–0.05	0.14	59.86	–59.72	0	–99.0%

Table 3
Results of numerical experiments for test projects from the set J90

Algorithm	t	F_{1000}	F_{5000}	F_M	F_A	$\#F_{best}$	$\%F_{best}$
Alg1	31.10	2.23	2.64	71.03	–68.39	2	–70.7%
Alg2	12.37	7.69	8.50	71.17	–62.67	567	–5.8%
Alg3	0.27	2.40	3.14	58.07	–54.93	7	–65.2%
Alg4	166.05	2.29	2.67	61.98	–59.30	21	–70.4%
Alg5	0.56	–3.26	–3.09	50.88	–53.96	0	–134.2%

The solution with the highest average value of F (for each of the sets J30, J60 and J90) was generated by backward serial SGS procedure with the optimisation of project contractual milestones completion time (Alg2).

The solution with the lowest value of F (for each of the sets J30, J60 and J90) was generated by backward parallel SGS procedure (Alg4 and Alg5). While using optimisation procedure for project contractual milestone deadlines (Alg4) improves the quality of the solutions obtained, they are still worse than those generated with a serial SGS procedure.

Forward scheduling with predefined resource allocation (Alg1) yields values of the objective function component F_M (determined based on the client's discounted payments CFM_m) similar to those generated by Alg2; however, the contractor's discounted expenditure (component F_A) is higher due to earlier starting time of some activities, whose right shifts in the Gantt chart proved infeasible with the predefined resource allocation [8–11].

The lowest discounted expenditure (component F_A) is recorded for backward scheduling with the predefined project milestone deadlines equal to the contractual milestone deadlines (Alg3 and Alg5). However, for solutions generated by Alg3 and Alg5, the value of the client's discounted payments (component F_M) is lower than for the schedules generated by the proposed Alg2 or Alg4 procedure.

While the procedure for the optimisation of project milestones completion time analysed herein is time consuming, it improves the solutions obtained.

7. Summary

In the paper, the problem of DCF maximising for a project settled by milestones was discussed from the contractor's perspective. The proposed model introduces a financial stage settlement of the project, which can be beneficial for both the contractor and the client. The financial optimization of project takes into account the expenses assigned to activities and client's payments for the completed stages of the work, less contractual penalties in case of delays.

In the schedule appropriate for the problem activities should be planned as late as possible but at the earliest stages completion of the project. A new backward scheduling procedure, generating such schedules, is developed. Numeric experiments were run, based on the simulated annealing metaheuristics using various algorithms of solution generation for the problem discussed. The results of experiments confirm the effectiveness of the procedure proposed herein.

The problem analysed is of current interest and may prove useful in practice. Future work will focus on the development of an algorithm that uses a justification technique dedicated to the problem being analysed and compares it to the procedure described in the paper.

References

[1] P. Brucker, A. Drexl, R. Mohring, K. Neumann, and E. Pesch, "Resource-constrained project scheduling: Notation, classification, models, and methods", *Eur. J. Oper. Res.*,112(1), 3–41 (1999).

[2] S. Hartmann and D. Briskorn, "A survey of variants and extensions of the resource-constrained project scheduling problem", *Eur. J. Oper. Res.*, 207(1), 1-14 (2012).

[3] R. Kolisch and R. Padman, "An integrated survey of deterministic project scheduling", *OMEGA*, 29, 249–272 (2001).

[4] A.H. Russell, "Cash flows in networks", *Manag. Sci.*, 16, 357–373 (1970).

[5] D.E. Smith-Daniels, R. Padman, and V.L. Smith-Daniels, "Heuristic scheduling of capital constrained projects", *J. Oper. Manag.*, 14(3), 241–254 (1996).

[6] A. Fink and J. Homberger, "An ant-based coordination mechanism for resource-constrained project scheduling with multiple agents and cash flow objectives", *Flex. Serv. Manuf. J.*, 25, 2013, 94–121 (2013).

[7] Y. Khoshjahan, A. Najafi, and B. Nadjafi, "Resource constrained project scheduling problem with discounted earliness-tardiness penalties: Mathematical modeling and solving procedure", *Comput. Ind. Eng.*, 66, 293–300 (2013).

[8] M. Klimek and P. Łebkowski, *Scheduling of a Project Settled by Milestones*, AGH, Cracow, 2015, [in Polish].

[9] M. Klimek and P. Łebkowski, "Robustness of schedules for project scheduling problem with cash flow optimisation", *Bull. Pol. Ac.: Tech.*, 61(4), 1005–1015 (2013).

[10] M. Klimek and P. Łebkowski, "Proactive project scheduling with cash flow maximisation", *Selected Conference Proceedings, MANUFACTURING*, Poznań, 2014, 72–86 (2016).

[11] M. Klimek and P. Łebkowski, "Heuristics for project scheduling with discounted cash flows optimisation", *Bull. Pol. Ac.: Tech.*, 63(3), 613–622 (2015).

[12] P. Leyman and M. Vanhoucke, "Payment models and net present value optimization for resource-constrained project scheduling", *Comput. Ind. Eng.*, 91, 139–153 (2016).

[13] S.S. Liu and C.J. Wang, "Resource-constrained construction project scheduling model for profit maximization considering cash flow", *Autom. Constr.*, 17(8), 966–974 (2008).

[14] M. Mika, G. Waligóra, and J. Węglarz, "Simulated annealing and tabu search for multi-mode resource-constrained project scheduling with positive discounted cash flows and different payment models", *Eur. J. Oper. Res.*, 164(3), 639–668 (2005).

[15] T. Selle and J. Zimmermann, "A bidirectional heuristic for maximizing the net present value of large-scale projects subject to limited resources", *Nav. Res. Logist.*, 50(2), 130–148 (2003).

[16] D. Thiruvady, M. Wallace, H. Gu, and A. Schutt, "A Lagrangian relaxation and ACO hybrid for resource constrained project scheduling with discounted cash flows", *J. Heuristics*, 20, 643–676 (2014).

[17] G. Ulusoy, E. Sivrikaya-Serifoglu, and S. Sahin, "Four payment models for the multi-mode resource constrained project scheduling problem with discounted cash flows", *Ann. Oper. Res.*, 102, 237–261 (2001).

[18] M. Vanhoucke, "A scatter search procedure for maximizing the net present value of a resource-constrained project with fixed activity cash flows", *Work. Paper* 2006/417, Gent, 1–23 (2006).

[19] M. Vanhoucke, E. Demeulemeester, and W. Herroelen, "On maximizing the net present value of a project under renewable resource constraints", *Manag. Sci.*, 47:1113–1121 (2001).

[20] M. Vanhoucke, E. Demeulemeester, and W. Herroelen, "Maximizing the net present value of a project with linear time-dependent cash flows", *Int. J. Prod. Res.*, 39(14), 3159–3181 (2001).

[21] W. Herroelen, B.D. Reyck, and E. Demeulemeester, "Project network models with discounted cash flows: A guided tour through recent developments", *Eur. J. Oper. Res.*, 100, 97–121 (1997).
[22] F. Bahrami and G. Moslehi, "Study of payment scheduling problem to achieve client-contractor agreement", *Int. J. Adv. Manuf. Tech.*, 64(1), 497–511 (2013).
[23] N. Dayanand and R. Padman, "Project contracts and payment schedules: The client's problem", *Manag. Sci.*, 47, 1654–1667 (2001).
[24] N. Dayanand and R. Padman, "On modelling payments in projects", *J. Oper. Res. Soc.*, 48, 906–918 (1997).
[25] G. Ulusoy and L. Özdamar, "A heuristic scheduling algorithm for improving the duration and net present value of a project", *Int. J. Oper. Prod. Manag.*, 15, 89–98 (1995).
[26] G. Ulusoy and S. Cebelli, "An equitable approach to the payment scheduling problem in project management", *Eur. J. Oper. Res.*, 127(2), 262–278 (2000).
[27] Z. He and Y. Xu, "Multi-mode project payment scheduling problems with bonus penalty structure", *Eur. J. Oper. Res.*, 189, 1191–1207 (2008).
[28] Z. He, N. Wang, T. Jia, and Y. Xu, "Simulated annealing and tabu search for multimode project payment scheduling", *Eur. J. Oper. Res.*, 198(3), 688–696 (2009).
[29] E. Goldratt, "The goal", New York: North River Press, (1992).
[30] E. Goldratt, "Critical chain", Great Barrington: The North River Press, (1997).
[31] M. Ranjbar, "An optimal NPV project scheduling with fixed work content and payment on milestones", *Int. J. Ind. Eng. Prod. Res.*, 22(3), 181–186 (2011).
[32] R. Kolisch and A. Sprecher, "PSPLIB – a project scheduling library", *Eur. J. Oper. Res.*, 96, 205–216 (1997).
[33] R. Kolisch, "Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation", *Eur. J. Oper. Res.*, 90, 320–333 (1996).
[34] S. Baroum and J. Patterson, "The development of cash flow weight procedures for maximizing the net present value of a project", *J. Oper. Manag.*, 14(3), 209–227 (1996).
[35] J.P. Pinder and A.S. Marucheck, "Using discounted cash flow heuristics to improve project net present value", *J. Oper. Manag.*, 14, 229–240 (1996).
[36] C. Schwindt, J. Zimmermann, "A steepest ascent approach to maximizing the net present value of projects", *Math. Methods Oper. Res.*, 53, 435–450 (2001).
[37] J. Goncalves, M. Resende, and J. Mendes J., "A biased random-key genetic algorithm with forward-backward improvement for the resource constrained project scheduling problem", *J. Heuristics*, 17(5), 467–486 (2011).
[38] R.K.-Y. Li and J. Willis, "An iterative scheduling technique for resource-constrained project scheduling", *Eur. J. Oper. Res.*, 56, 370–379 (1992).
[39] P. Tormos and A. Lova, "A competitive heuristic solution technique for resource-constrained project scheduling", *Ann. Oper. Res.*, 102, 65–81 (2001).
[40] P. Tormos and A. Lova, "An efficient multi-pass heuristic for project scheduling with constrained resources", *Int. J. Prod. Res.*, 41(5), 1071–1086 (2003).
[41] V. Valls, F. Ballestin, and S. Quintanilla S., "Justification and RCPSP: a technique that pays", *Eur. J. Oper. Res.*, 165(2), 375–86 (2005).
[42] V. Valls, F. Ballestin, and S. Quintanilla S., "Justification technique generalisations", In: J. Józefowska and J. Węglarz (eds.): *Perspectives in Modern Project Scheduling*, Springer, Berlin, 205-223 (2006).
[43] V. Valls, F. Ballestin, and S. Quintanilla S., "A hybrid genetic algorithm for the RCPSP", *Eur. J. Oper. Res.*, 185(2), pp. 495–508 (2008).
[44] J.D. Wiest, "Some properties of schedules for large projects with limited resources", *Oper. Res.*, 12 (3), 395–418 (1964).
[45] J. Józefowska and J. Węglarz (eds.), *Perspectives in modern project scheduling*, Springer, Berlin, 2006.
[46] J. Błażewicz, J.K. Lenstra, and A.R. Kan, "Scheduling subject to resource constraints – classification and complexity", *Discret. Appl. Math.*, 5, 11–24 (1983).
[47] S. Hartmann and R. Kolisch, "Experimental evaluation of state-of-the-art heuristics for the resource-constrained project scheduling problem", *Eur. J. Oper. Res.*, 127, 394–407 (2000).
[48] R. Kolisch and S. Hartmann, "Experimental Investigation of Heuristics for Resource-Constrained Project Scheduling: An Update", *Eur. J. Oper. Res.*, 74(1), 23–37 (2006).
[49] K. Bouleimen and H. Lecocq, "A new efficient simulated annealing algorithm for the resource constrained project scheduling problem and its multiple version", *Eur. J. Oper. Res.*, 149, 268–281 (2003).
[50] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi, "Optimization by simulated annealing", *Science*, 220, 671–680 (1983).