Management and Production Engineering Review
Volume 9 • Number 4 • December 2018 • pp. 106–116
DOI: 10.24425/119551

INSERTION ALGORITHMS TO SOLVE THE RESOURCE-CONSTRAINED MULTI-STAGE PROJECT SCHEDULING PROBLEM WITH DISCOUNTED CASH FLOW MAXIMIZATION

Marcin Klimek

State School of Higher Education, Department of Computer Science, Poland

Corresponding author:
Marcin Klimek
State School of Higher Education
Department of Computer Science
Sidorska 95-97, Biala Podlaska, Poland
phone: (+48) 83 344 99 00
e-mail: m.klimek@dydaktyka.pswbp.pl

Received: 25 October 2017
Accepted: 12 June 2018

Abstract
The article presents the problem of scheduling a multi-stage project with limited availability of resources with the discounted cash flow maximization criterion from the perspective of a contractor. The contractor's cash outflows are associated with the execution of activities. The client's payments (cash inflows for the contractor) are performed after completing the agreed project's stages. The proposed solution for this problem is the use of insertion algorithms. Schedules are generated using forward and backward schedule generation schemes and modified justification techniques. The effectiveness of the proposed procedures is the subject of the examination with the use of standard test instances with additionally defined financial settlements of a project.

Keywords
insertion algorithms, discounted cash flow, resource-constrained project scheduling, priority rules.

Introduction

Resource-Constrained Project Scheduling Problem (RCPSP) has been one of the significant optimization problems. For the need of such a problem, there are analyzed various types of resources, ways of performing activities, optimization criteria, etc. [1–3]. One of the most important aspects of planning a project in practice is its financial optimization that takes into account all cash flows associated with the project. In the majority of research works cash flows are discounted, which means that their NPV (Net Present Value) is calculated. This examination includes maximization models with discounted cash flows RCPSP-DC (RCPSP with Discounted Cash Flows) and scheduling project payments PPS (Payment Project Scheduling) [4–15].

Payment problems are the subject of examination from the perspective of a contractor and/or a customer. There are determined, for instance, the amount of customer's particular payments and total payments for completing the project, the number of tranches, the deadlines, etc. There has been a search for a solution satisfactory for the customer [5–6], the contractor [7–9, 11–15] or for both parties [4, 10].

This study contains the analysis of the financial optimization of the project from the perspective of a contractor. Taking into account the contractor's point of view, cash flows in PPS problems consist of cash inflows, e.g. the contractor's expenses related to the realization of the activity with the use of resources as well as cash outflows, e.g. the customer's payments for the contractor for the completed project or its stages. The customer may make payments [7–9] one at a time, after the end of the project (the LSP model – Lump-Sum Payment) or many times: after specific events (the PEO model – Payments at Event Occurrences), i.e. the completion

of the project's stage or the completion of activities (the PAC model – Payments at Activities' Completion times), or in fixed time intervals (the ETI model – Equal Time Intervals, the PP model – Progress Payments).

The bonus-penalty system [16] is also introduced when settling the project, e.g. penalties for exceeding the agreed deadline for completing the project (a stage), benefits for an earlier completion of the project (a stage), etc. Penalties and bonuses are introduced to stimulate the contractor to complete the project activities as quick as possible, on time and to compensate for any possible losses for the customer caused by delays in completing the project. The bonus-penalty system is effective from the perspective of a contractor when the benefits from awards are higher than the contractor's costs borne in connection with a faster completion of activities (stages), and the penalties for the lack of punctuality are higher than the benefits from the later completion of activities (stages). The bonus-penalty system is effective from the perspective of the customer when the contractor's bonus for a quicker completion of activities (stages) does not exceed the customer's benefits from a quicker completion of activities (stages), and the penalty for the untimely completion of activities is higher than the customer's lost benefits from the timely completion of activities (stages).

This study proposes the author's model [11–15], for the maximization of the amounts of discounted cash flows related to the project from the perspective of the contractor, with stage settlements in which there are agreed stages of the project (milestones) with deadlines as well as amounts of payments for their completion. Penalties reducing the customer's stage payments are calculated in the case of a delayed completion of the project's stages. Benefits from a stage system of settlements for the contractor include the possibility to receive earlier payments for performed activities that may be allocated for current operations (completion of new activities, purchase of materials, etc.). Earlier payments, made before the completion of activities, are unfavorable for the customer but the proposed settlement model gives the customer the possibility to exercise a better control over the course of the project, and the introduction of penalties provides an additional stimulus for the contractor to complete the project's stages.

The proposed model of multi-stage settlements may be useful in practice. The milestone technique is an important element of planning activities in course of completing projects, e.g. it is used to determine the degree of the project's completion, it facilitates project management by increasing the possibility to control the course and the timely completion of the project, etc. The number of milestones should not be large to emphasize their "exceptional" nature. Apart from the author's works, multi-stage project settlements have not been examined in this form in research concerning RCPSP. Cash flows related to the project's stages are analyzed for the Discrete Time/Cost Trade-off Problem (DTCTP) [16], Multi-Mode RCPSP (MMRCPSP) [17, 18], and this study examines the Single-Mode RCPSP.

In previous works concerning the model analyzed here, the robust scheduling [11, 12] and project planning under uncertainty [15] were taken into consideration. There were used the simulated annealing algorithms [12, 13], with the generation of the following solutions techniques: backward scheduling with completion times optimization of the project stages [12, 14] or forward scheduling with activity right-shifts procedure [13].

This article presents new insertion algorithms dedicated to solve the proposed, deterministic problem concerning financial optimization of a multi-stage project. They are designed similarly to effective procedures for a flow shop. The purpose of the study is to analyze the effectiveness of these insertion algorithms for the examined problem as well as the used priority rules and dedicated techniques for generating solutions, especially justification technique, adapted to the problem. The effectiveness is tested for problems from the PSPLIB (Project Scheduling Problem LIBrary) [19] with additionally defined contractual stages of the project and specific cash flows for the financial settlements of activities [14].

Problem formulation

The object of the analysis is a nonpreemptive, single-mode RCPSP in which the project is presented in the AON representation (Activity-On-Node) as a directed graph G(V, E) in which V is the set of nodes corresponding to activities, and E is the set of arcs describing precedence relations between finish-start zero-lag precedence activities. Activities are executed with the use of renewable resources the number of which is limited and fixed in time.

The objective of scheduling is to find a schedule (starting times for activities and, on this basis, financial settlements) with the maximum amount of discounted cash flows from the perspective of the contractor, with expenses related to the execution of activities as well as with revenues earned on ac-

count of the completion of contractual stages of the project

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

with the following resources constraints (2), order constraints (3) and contractual settlements (4) and (5):

$$\sum_{i \in J(t)} r_{ik} \leq a_k, \quad \forall t : t = 1, ST_{N_A+1}, \forall k : k = 1, ..., K, \quad (2)$$

$$ST_i + d_i \leq ST_j, \qquad \forall (i,j) \in E, \quad (3)$$

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

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

where F – the objective function, sum of discounted cash flows, N_A – the number of activities in the project, N_M – the number of contractual stages defined in the project, i – index of activities, $i = 1, \ldots, N_A$, m – index of project stage, $m = 1, \ldots, N_M$, CFA_i – contractor's expenses related to the completion of activity i, α – discount rate, K – the number of types of renewable resources, r_{ik} – demand for activities and resource type $k = 1, \ldots, K$, a_k – availability of resources type k, ST_i – starting time of activity i, d_i – duration of activity i, $J(t)$ – set of activities executed in period $[t-1, t]$, FT_i – completion time of activity i ($FT_i = ST_i + d_i$), CFM_m – customer's payments for the completion of the m stage of the project determined for the planned schedule, MT_m – the completion time of the m stage of the project in the planned schedule, MA_m – set of activities to be completed in the m stage of the project, MP_m – customer's payment for the completion of the m stage of the project, MD_m – agreed completion deadline of the m stage of the project, MC_m – agreed unit penalty for exceeding the deadline MD_m for the completion of the m stage of the project.

The financial settlements of the project from the perspective of the contractor assume that the expenses (cash outflows) include the costs of completing the activities CFA_i (for activities $m = 1, \ldots, N_A$), and cash inflows include the customer's payments CFM_m (for stages $m = 1, \ldots, N_M$) for the completed stages of project.

It has been assumed that all the contractor's expenses may be linked with the completion of particular activities. CFA_i expenses are calculated e.g. on the basis of costs of investing resources, the employees' remuneration, consumed materials, their transportation, etc.

The customer's stage payments CFM_m (4) are determined as the difference between the agreed amount of payments MP_m and the possible penalty for exceeding the completion deadline MD_m calculated while taking into account the agreed unit cost of delays MC_m. If stages are completed earlier than within the deadline MD_m, the contractor receives financial resources earlier, with a greater discounted value.

There has been made the assumption that the customer performs CFM_m payments at the exact time of the completion of the project stages MT_m planned in the current schedule and CFA_i expenses are borne by the contractor in planned dates of starting the activities. This study does not contain the analysis of the problem of delays in payments.

Proposed insertion algorithms

The project scheduling problem with limited resources, as the generalization of the job shop problem, is a strongly NP difficult problem [20] for which it is reasonable to use effective heuristic algorithms. The analysis of the effectiveness of algorithms for RCPSP may be found in overview studies [21, 22]. Research works present primarily approximate (heuristic) algorithms with polynomial calculation complexity, which find the schedules in an acceptable time, also for problems with a greater number of activities.

This study uses construction algorithms based on the insertion method that will build schedules on the basis of the principles for inserting subsequent activities to the schedule. They are used to quickly generate schedules for projects consisting of a large number of activities as well as to create inaugural, initial solutions which are improved (the quality of solutions for construction algorithms may be unacceptable) by metaheuristics, namely simulated annealing, genetic algorithms, etc.

The insertion algorithms are used for various optimization problems, including scheduling problems. Insertion algorithms for scheduling issues most often consist of two stages:

1. the initial stage in which the initial activity list is determined by using the chosen algorithm applying priority rules,
2. the basic stage in which the n sequence of partial permutations is generated, starting with a single-element permutation and ending with an n-element permutation. Each subsequent partial

permutation is created taking into account the previous permutation and the subsequent activity inserted from the initial list.

The initial activity list, the sequence of drawing activities from it as well as the positions attributed to activities in partial permutations, are specific for particular insertion algorithms.

This article proposes the following insertion algorithms **Alg1**, **Alg2** and **Alg3**, developed by the author for RCPSP, based on the concepts of procedures used for other scheduling problems, i.e. the permutation flow shop [23, 24].

The **Alg1** algorithm proceeds in the following steps [25]:

Step 1:

Creating the initial list L consisting of all the activities which may be started at time $t = 0$, arranged on the basis of the adopted priority rule.

Step 2:

Inserting the first activity from the L list to all possible positions on the current activity list P (taking into account order constraints) and generating a partial schedule in each case (for each insertion position). From among all possible partial schedules there is chosen the schedule for which the best solution evaluated with the adopted optimization criterion of partial schedules is determined.

Step 3:

Updating the L list: removing the activity inserted in step 2 and inserting all consequents of this activity, which all predecessors are already included in the P list, observing the order of activities on the L list resulting from the adopted priority rule.

Steps 2–3 are performed until all activities are placed on the L list from which a schedule being the solution to the RCPSP problem is generated with the use of SGS.

In the description of the algorithm:

- P list – the activity list from which the current schedule is created with the use of the SGS decoding procedure,
- L list – the list of currently available activities that may be started, which means that its predecessors are activities already placed on the P list).

Procedures of **Alg2** and **Alg3** act in a similar manner as **Alg1**. The order of activities on the L list for **Alg2** and **Alg3** has no effect on the course of the algorithm (the order may be any). Step 2 of the **Alg2** procedure includes the sample of insertion of each activity from the L list into the last position on the P list and choosing such activity for this position for which the best partial schedule is generated. In turn, step 2 of the **Alg3** procedure includes the sample of insertion of each activity from the L list to all possible positions on the P list and choosing such activity and such insertion position for which the best partial schedule is generated.

Step 2 of the **Alg1**, **Alg2**, **Alg3** algorithms involves the evaluation of partial schedules. There is determined the value of the F function for the full schedule found for the activity list consisting of activities found on the current P list and then from the remaining project activities in the order resulting from the applied priority rule. Priority rules applied in calculation experiments are as follows:

- R0 – random priorities of activities,
- R1 – the minimum latest starting time of the activity taking into account the agreed deadlines for the completion of project stages,
- R2 – the minimum latest finish time of the activity taking into account the agreed deadlines for the completion of project stages,
- R3 – the maximum number of all the successors of the activities,
- R4 – the maximum sum of the duration periods of a given activity and all of its successors,
- R5 – the minimum number of a project stage in which the activity is executed and, with the same stages, the minimum cost of executing activities (the CFA_i).

These rules are used to arrange activities that have not been placed on the P list yet (in determining the F objective function) as well as to arrange activities on the L list in the **Alg1** procedure. If the activities have the same priorities for a given priority rule, activities marked with a lower number are placed on earlier positions of the list.

Solutions in the proposed insertion algorithms are stored in indirect representation, the so-called activity list, which is the permutation of the numbers of subsequent activities taking into account precedence relations. The activity list is transformed to direct representation, e.g. a vector of starting times for activities, with the use of Schedule Generation Schemes SGS, which generate feasible schedules meeting the order and resource constraints. In this study, there are frequently used the following decoding procedures for RCPSP: serial SGS and parallel SGS [26]. The schedule may be determined with the use of SGS as a result of planning subsequent activities from the beginning of the activity list (forward scheduling) or from the end of the activity list, determining the times of starting the activities with the agreed due date (backward scheduling).

The forward or backward schedule determined with the use of SGS procedures may be improved for the analyzed problem of the financial optimization of a multi-stage project similarly as for RCPSP-DC. It

is recommended to acquire the customer's payments for the completed stages of project as quickly as possible and to bear expenses related to commenced activities as late as possible. The growth in the sum of accumulated cash flows F is always achieved with a later start of activities for which the completion times of the project's stages are not changed. The completion of the project's stages earlier than within the agreed deadlines may be beneficial due to a higher NPV of the customer's earlier payments.

The known techniques of generating solutions for RCPSP-DC, namely the bidirectional SGS procedure [27], right and left shift algorithms for activities, etc., the overview of which may be found in the study of M. Vanhoucke [28], do not build solutions relevant for the analyzed optimization model. As a result, procedures developed by the author and dedicated for the examined problem are proposed to improve the solutions determined with the use of SGS:

- backward scheduling with the optimization (shifting) of the completion time of the agreed stages of the project [12, 14],
- the techniques of right justification by extremes and left justification by extremes taking into account the completion time of the agreed project stages.

The operation of both of the above-mentioned procedures of improving schedules is explained for the illustrative example.

Illustrative example

Let us present an illustrative example to explain the proposed problem and the techniques of generating solutions. The AON activity network for an exemplary project with agreed settlements is presented on Fig. 1.

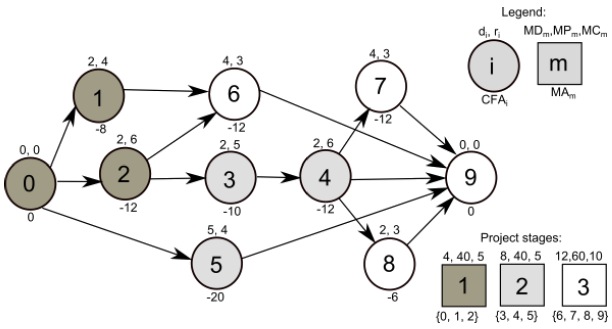

Fig. 1. An example project with agreed stages in AON representation.

The project consists of eight activities (activities 0 and 9 are dummy activities representing, respectively the initial and the final node in the graph G(V, E)) executed with the use of one type of resource with availability equal to 10. The project defines three agreed stages of settlements, in which activities $MA_1 = \{0, 1, 2\}$, $MA_2 = \{3, 4, 5\}$, $MA_3 = \{6, 7, 8, 9\}$ are executed with agreed deadlines $MD_1 = 4$, $MD_2 = 8$, $MD_3 = 12$. The customer's stage payments are $MP_1 = 40$, $MP_2 = 40$, $MP_3 = 60$ and they may be reduced by the costs of possible delays in the execution of project stages calculated on the basis of unit costs $MC_1 = 5$, $MC_2 = 5$, $MC_3 = 10$. The discount rate $\alpha = 0.01$ was adopted when determining F.

There is the need to explain the manner of generating dedicated solutions on the basis of the activity list for the problem of the financial optimization of a multi-stage project. Let us assume that the activity list {1, 5, 2, 3, 6, 4, 7, 8} is to be decoded.

The first proposed procedure for generating solutions is backward scheduling with the optimization of the completion time of the agreed stages of the project. This procedure uses SGS with backward planning for the adopted completion time of the project's stages. Subsequent iterations include unit shifts to the left of the completion time of all stages of the project MT_m, starting with the first and ending with the last. They are introduced as long as this operation increases the value of the$F$ objective function. The course of the optimization of the agreed stages' completion time for the analyzed project is illustrated in Figs. 2–5.

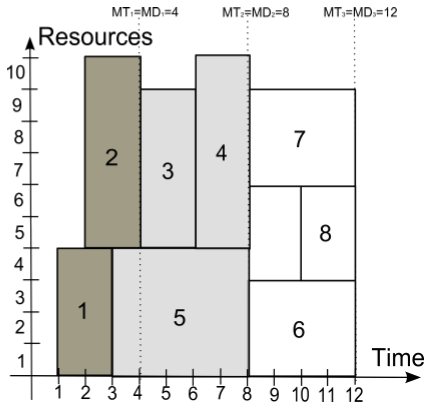

Fig. 2. Backward schedule generated with the use of serial SGS for activity list {1, 5, 2, 3, 6, 4, 7, 8} with adopted initial, agreed completion time of stages.

The backward schedule determined for the analyzed list {1, 5, 2, 3, 6, 4, 7, 8} and serial SGS with the completion time of stages such as the agreed $MT_1 = MD_1 = 4$, $MT_2 = MD_2 = 8$, $MT_3 = MT_1 = 12$ is presented in Fig. 2 (value of the objective function for this schedule $F = 40.98$). The procedure for left shifts of the project's stages starts from shifting the completion time of the first stage. Starting SGS assuming the subsequent unit shifts:

- $MT_1 = 3$ – increases the value of the objective function F from 40.98 to 41.25,
- $MT_1 = 2$ – increases the value of the objective function F from 41.25 to 41.45,
- $MT_1 = 1$ – it is impossible to create a schedule, the procedure proceeds to a left shift of the second stage of the project.

The schedule determined after shifting the first stage from $F = 41.45$ is presented in Fig. 3.

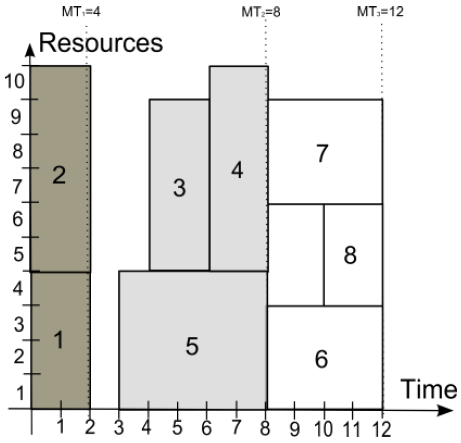

Fig. 3. Backward schedule determined on the basis of the optimization for completion time of the first stage.

The unit shift of the completion time $MT_2 = 7$ in the second stage reduces the value of the F objective function from 41.45 to 41.41 (the growth in the customer's discounted payment for the second stage is smaller than the growth in the amount of the contractor's discounted expenses borne on account of an earlier completion of activities 3, 4, 5). The procedure proceeds to a left shift of the third stage of the project.

In the third stage, assuming the completion time of the stage shifted by a unit $MT_3 = 11$, the value of the F objective function increases from 41.45 to 41.49. Assuming the completion time of the stage shifted by another unit $MT_3 = 10$, the value of the F objective function increases from 41.49 to 41.54 as a result of the increased discounted payment for the third stage. It is impossible to generate a feasible schedule within $MT_3 = 9$.

The schedule generated after the first passage of shifts in the stages of the project, with $F = 41.54$, is presented in Fig. 4.

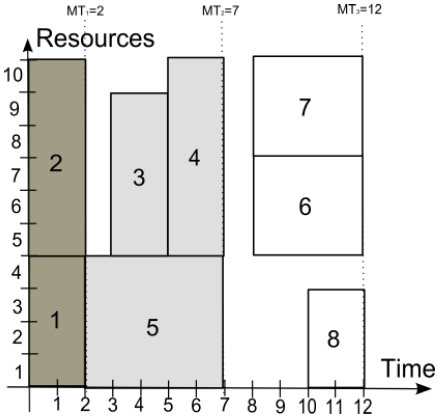

Fig. 4. Backward schedule determined on the basis of a single course of optimization for all completion time of the project's stages.

As a result of the optimization of the completion time of agreed stages, the value of the Fobjective function was improved from 40.98 to 41.54. The algorithm starts the improvement from the first stage again. It is impossible to shift the first stage with $MT_1 = 2$. Shifting the second stage to $MT_2 = 7$ causes the growth of F from 41.54 to 41.72. With $MT_2 = 6$ it is impossible to build a feasible schedule, as with shifting the third stage to $MT_3 = 9$. The schedule with $F = 41.72$, presented in Fig. 5, is created as a consequence of the optimization of the completion time of the project's stages. The further shift of stages does not bring any growth in the F objective function and the procedure ends.

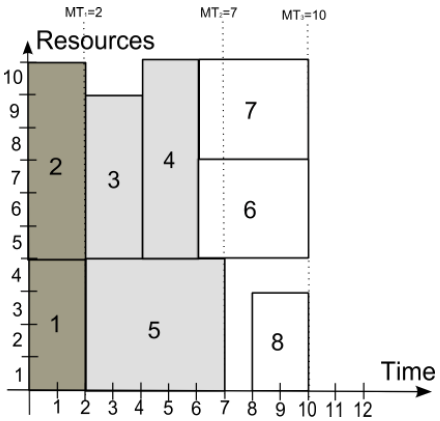

Fig. 5. Final backward schedule determined as a result of the application of the optimization procedure for all completion time of the project's stages.

Solutions relevant for the analyzed problem of the financial optimization of a multi-stage project

may also be generated with the use of justification techniques known for RCPSP [29, 30]. Right justification RJ and left justification LJ are often used in combination. The justification of a given activity to the right (to the left) consists in determining the latest (the earliest) possible starting time for this activity, taking into account the order and resource constraints as well as the current schedule. For the analyzed problem with RJ, activities are shifted so as not to change the current deadlines for the completion of the project stages. Activities are to be started as late as possible, while the agreed project stages are to be completed as early as possible.

The analysis of justification techniques for exemplary schedules has shown that it may be effective to use triple justification RJ+LJ+RJ. Let us assume that there takes place the revision of the schedule from Fig. 6, generated forward with the use of serial SGS for the activity list {1, 5, 2, 3, 6, 4, 7, 8} with the value of the F objective function = 39.87. The subsequent transformations of the schedule with the use of triple justification RJ + LJ + RJ are presented in Figs. 7–9.

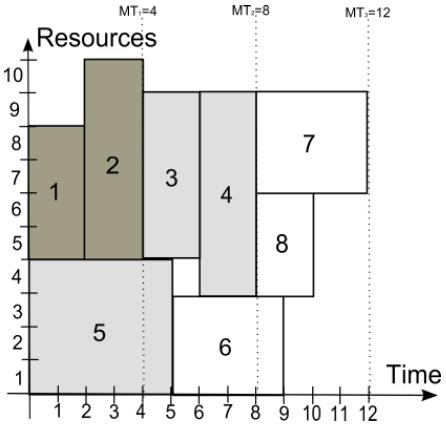

Fig. 6. Forward schedule generated with the use of a serial SGS for the activity list {1, 5, 2, 3, 6, 4, 7, 8}.

At the beginning there is used right justification in which activities with the maximum completion time are subsequently chosen (RJ by extremes) – activities 7, 8, 6, 4, 3, 5, 2, 1. The use of RJ generates the schedule with a better quality of $F = 40.99$, presented in Fig. 7.

Left justification is performed after the application of RJ. Subsequent activities with the minimum starting time (LJ by extremes) are chosen for LJ in the schedule generated after RJ from Fig. 7 – activities 1, 2, 5, 3, 4, 6, 7, 8. As a result of LJ, there is created the schedule presented in Fig. 8 with a higher value of the F objective function = 41.66, due to an earlier completion of all stages of the project.

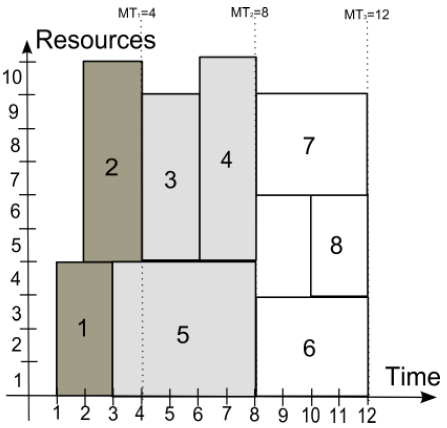

Fig. 7. Schedule after RJ for $MT_1 = 4$, $MT_2 = 8$, $MT_3 = 12$.

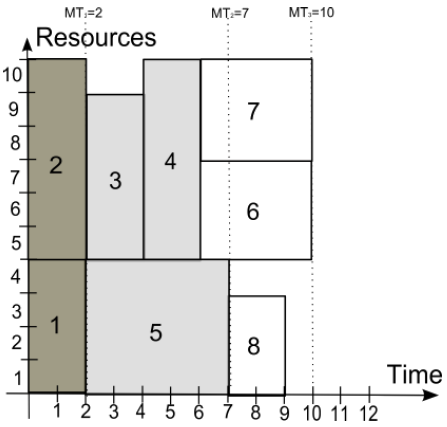

Fig. 8. Schedule after RJ + LJ.

RJ is performed for the schedule from Fig. 8, assuming the shifted completion time of the agreed project's stages ($MT_1 = 2$, $MT_2 = 7$, $MT_3 = 10$) for subsequently justified activities 7, 6, 8, 5, 4, 3, 2, 1. There is generated a new schedule, presented in Fig. 9, with a higher value of the F objective function $F = 41.72$ than the schedule from Fig. 8, due to a later start of activity 8.

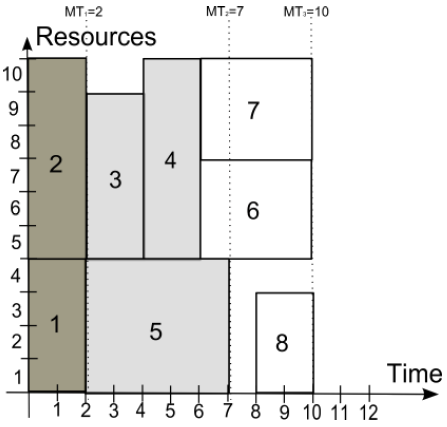

Fig. 9. Schedule after RJ + LJ + RJ.

The schedule from Fig. 9 is the final solution to the problem obtained with the use of justification RJ + LJ + RJ for the activity list {1, 5, 2, 3, 6, 4, 7, 8}. It is identical as the schedule from Fig. 5, found with the use of the backward scheduling procedure with the optimization of the completion time of the project's stages.

Results of experiments

The experiments were performed using an application implemented in the C# language in Visual Studio.NET, using a computer with an Intel Core i7-4770 processor CPU 3.4 GHz, 8 GB RAM, for 480 test instances from the set J30 (30-activity problems) and for 480 instances from the set J90 (90-activity problems) from the PSPLIB [19].

The agreed settlements are defined for each project from the PSPLIB on the basis of a baseline schedule S which is created as a result of the application of a serial procedure SGS for the activity list $\{1, 2, \ldots, 30\}$ for the set J30 or $\{1, 2, \ldots, 90\}$ for the set J90. There are defined three project stages with deadlines $MD_1 = T/3$, $MD_2 = 2T/3$ and $MD_3 = T$ (T – duration of the project in the base schedule S). The sets of stage activities MA_m are determined on the basis of the S schedule:

- the set MA_1 contains all activities the completion time of which is lower than or equal to MD_1,
- the set MA_2 contains all activities the completion time of which is lower than or equal to MD_2 and higher than MD_1,
- the set MA_3 contains the remaining activities.

In the financial settlements for each test instance: $MP_1 = 60$, $MP_2 = 60$, $MP_3 = 120$, $MC_1 = 1.5$, $MC_2 = 1.5$, $MC_3 = 3$, CFA_i costs are determined as costs proportional to the total demand for resources and the completion time of a given activity assuming that their sum for all activities is 100 [12]. The discount rate adopted in the experiments is amounted to $\alpha = 0.01$.

The purpose of the experiments was to evaluate the effectiveness of the developed insertion algorithms for the analyzed problem as well as to find the best priority rules (among the rules R0–R5) and the techniques of generating solutions. For the need of comparative purposes, at the beginning there were performed computational experiments for a single-pass priority rule heuristic as well as for the applied priority rules and techniques of generating solutions. The results of experiments for the single-pass priority are presented in Table 1, while Tables 2 and 3 contain the results for the insertion algorithms **Alg1**, **Alg2** and **Alg3**. The calculation time and the number of tested solutions for **Alg1**, **Alg2** and **Alg3** are presented in Table 4.

The **Alg1** algorithm is the most effective one. It uses the backward scheduling procedure with the optimization of the completion time of the project's stages, with the improvement of solutions using the left and then the right justification. The best results are achieved for the priority rule R2 (the minimum latest finish time) which proved to be the most effective one for the projects from both the set J30 and the set J90.

The use of effective priority rules improves the quality of the obtained solutions. Schedules generated for algorithms using the rule with random priorities of activities R0 are of worse quality – the lowest average value of the F objective function. R1, R2 rules are effective priority rules.

A considerable improvement in the quality of solutions is always observed after the application of the triple justification (for forward scheduling) or the double justification of solutions (backward scheduling). The use of justification increases not only the calculation time but also the value of the F objective function. The most effective technique of generating solutions is backward scheduling with the optimization of the completion time of the project's stages, with the improvement of solutions using the left and then the right justification.

Table 1
Results of experiments for single pass priority rule algorithms.

	J30				J90			
	Forward scheduling		Backward scheduling		Forward scheduling		Backward scheduling	
	RJ	RJ + LJ + RJ	–	LJ + RJ	RJ	RJ + LJ + RJ	–	LJ + RJ
R0	54.73	66.75	63.26	69.47	19.78	39.85	19.78	39.85
R1	67.02	71.41	69.31	72.26	36.58	46.17	36.58	46.17
R2	69.57	73.05	69.72	72.51	37.78	46.79	37.78	46.79
R3	65.67	71.49	68.79	71.71	29.41	43.18	29.41	43.18
R4	63.44	70.27	67.88	71.59	27.43	42.90	27.43	42.90
R5	64.11	69.83	67.63	71.19	31.55	44.60	31.55	44.60

Table 2
Results of experiments for projects consisting of 30 activities – proposed insertion algorithms.

		Average objective function F				Number of the best solutions			
		Forward scheduling		Backward scheduling		Forward scheduling		Backward scheduling	
		RJ	RJ + LJ + RJ	–	LJ + RJ	RJ	RJ + LJ + RJ	–	LJ + RJ
Alg1	R0	31.36	45.39	39.34	50.18	103	133	83	148
	R1	67.89	73.10	75.56	77.04	178	204	149	206
	R2	76.62	77.12	75.88	77.32	182	209	189	235
	R3	76.60	77.00	75.76	77.10	142	159	139	198
	R4	74.26	75.33	74.03	76.73	138	161	125	186
	R5	73.32	74.63	73.18	76.41	119	161	151	188
Alg2	R0	73.80	76.42	75.33	76.61	126	156	137	193
	R1	72.21	75.41	75.30	76.54	145	185	141	193
	R2	75.48	76.55	75.40	76.93	150	188	148	195
	R3	75.86	76.77	75.49	76.97	144	186	144	195
	R4	75.31	76.43	75.39	76.85	136	175	142	193
	R5	74.52	76.13	75.45	76.86	144	168	139	206
Alg3	R0	74.57	76.29	75.42	76.82	124	174	156	204
	R1	73.05	75.79	75.64	76.86	155	197	160	201
	R2	75.86	76.79	75.90	77.02	158	195	162	206
	R3	76.12	76.94	75.99	77.06	147	193	157	205
	R4	75.54	76.56	75.77	76.98	140	185	158	203
	R5	74.86	76.31	75.92	76.98	153	176	159	218

Table 3
Results of experiments for projects consisting of 90 activities – proposed insertion algorithms.

		Average objective function F				Number of the best solutions			
		Forward scheduling		Backward scheduling		Forward scheduling		Backward scheduling	
		RJ	RJ + LJ + RJ	–	LJ + RJ	RJ	RJ + LJ + RJ	–	LJ + RJ
Alg1	R0	31.36	45.39	39.34	50.18	22	26	7	28
	R1	51.34	51.94	47.84	52.21	36	50	28	52
	R2	51.19	52.05	49.57	52.57	35	49	42	68
	R3	41.82	46.07	39.80	51.59	23	30	17	40
	R4	40.62	44.99	38.38	51.43	24	27	17	41
	R5	43.60	51.55	48.71	51.86	23	30	19	47
Alg2	R0	41.25	49.45	48.37	51.94	24	39	17	44
	R1	48.34	52.04	48.64	52.26	24	42	14	53
	R2	48.86	52.22	49.04	52.38	24	37	14	56
	R3	45.22	51.01	48.88	52.21	23	33	13	46
	R4	44.49	50.85	48.65	52.12	23	30	8	44
	R5	46.72	51.70	48.54	52.18	24	31	8	50
Alg3	R0	42.22	49.57	49.17	52.04	22	36	17	50
	R1	48.77	52.04	49.56	52.41	25	46	19	62
	R2	49.22	52.27	49.81	52.46	24	48	17	62
	R3	45.61	51.24	49.67	52.31	24	34	15	53
	R4	45.18	51.20	49.50	52.20	24	35	11	53
	R5	47.14	51.87	49.50	52.30	24	34	13	65

The **Alg2** algorithm is the quickest one. It determines and checks the smallest number of schedules in the course of its activity. The slowest algorithm is **Alg3** that determines and checks the highest number of schedules in the course of its activity.

The number of schedules analyzed in insertion algorithms increases rapidly along with the increase in the size of the problem, which may prevent their application for very large projects. Several times more solutions are tested for projects consisting of 90 activities than for projects consisting of 30 activities.

Table 4
Comparison of calculation times and the number of verified schedules for analyzed insertion algorithms.

		Forward scheduling				Backward scheduling			
		RJ		RJ + LJ + RJ		–		LJ + RJ	
		CPU	No. sol.	CPU	No. sol.	CPU	No. sol.	CPU	No. sol.
J30	**Alg1**	0.006	157.9	0.068	158.0	0.035	158.1	0.092	158.1
	Alg2	0.004	124.2	0.055	123.1	0.028	124.1	0.075	123.2
	Alg3	0.009	217.7	0.096	215.3	0.048	217.1	0.127	215.5
J90	**Alg1**	0.043	1078.2	0.732	1078.1	0.397	1079.9	1.085	1079.1
	Alg2	0.030	711.7	0.499	699.4	0.256	702.1	0.715	695.0
	Alg3	0.056	1332.6	0.935	1308.3	0.476	1324.0	1.336	1313.4

Summary

The article proposes insertion algorithms for the problem of scheduling a multi-stage project with limited resources with the maximization criterion of the sum of discounted cash flows (cash outflows are assigned to activities and client's payments are realized for the completed stages of the project). These procedures are based on concepts used for the flow shop problem. Solutions are generated using modified justification techniques taking into account the special nature of the examined problem. In the schedule appropriate for the problem, activities should be planned as late as possible but with the earliest possible completion of the agreed project's stages.

The effectiveness of algorithms was tested for test instances from the PSPLIB with additionally specified financial settlements of the project's stages. The experiments have demonstrated a good efficiency of the **Alg1** algorithm and the backward scheduling procedure with the optimization of the completion time of the project's stages, with the improvement of solutions using the left and then the right justification.

Schedules generated by the proposed insertion algorithms will be used as inaugural solutions in the algorithms for local searches, i.e. simulated annealing, the development of which will be the subject of further work of the author. The subject matter is up-to-date and the proposed settlement model of a multi-stage project may be useful when executing practical projects.

References

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

[2] Hartmann S., Briskorn D., *A Survey of Variants and Extensions of the Resource-Constrained Project Scheduling Problem*, Eur. J. Oper. Res., 207, 1, 1–14, 2012.

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

[4] Bahrami F., Moslehi G., *Study of payment scheduling problem to achieve client-contractor agreement*, Int. J. Adv. Manuf. Tech., 64, 1, 497–511, 2013.

[5] Dayanand N., Padman R., *Project contracts and payment schedules: the client's problem*, Manag. Sci., 47, 1654–1667, 2001.

[6] Dayanand N., Padman R., *On modelling payments in projects*, J. Oper. Res. Soc., 48, 906–918, 1997.

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

[8] Mika M., Waligóra G., Węglarz J., *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.

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

[10] Ulusoy G., Cebelli S., *An equitable approach to the payment scheduling problem in project management*, Eur. J. Oper. Res., 127, 2, 262–278, 2000.

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

[12] Klimek M., Łebkowski P., *Scheduling of a project settled by milestones* [in Polish: *Harmonogramowanie projektu rozliczanego etapowo*, AGH, Cracow, 2015.

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

[14] Klimek M., Łebkowski P., *Financial optimisation of the scheduling for the multi-stage project*, Bull. Pol. Ac.: Tech., 65, 6, 899-908, 2017.

[15] Klimek M., Łebkowski P., *Cash flow maximization for project scheduling problem under uncertainty* [in Polish: *Maksymalizacja przepływów pieniężnych dla problemu harmonogramowania projektu w warunkach niepewności*], [in:] Innowacje w zarządzaniu i inżynierii produkcji, Vol. 1, Ryszard Knosala [Ed.], PTZP, Opole, pp. 562–577, 2017.

[16] Ranjbar M., *An optimal NPV project scheduling with fixed work content and payment on milestones*, Int. J. Ind. Eng. Prod. Res., 22, 3, 181–186, 2011.

[17] He Z., Xu Y., *Multi-mode project payment scheduling problems with bonus penalty structure*, 189, 1191–1207, 2008.

[18] He Z., Wang N., Jia T., Xu Y., *Simulated annealing and tabu search for multimode project payment scheduling*, Eur. J. Oper. Res., 198, 3, 688–696, 2009.

[19] Kolisch R., Sprecher A., *PSPLIB – a project scheduling library*, Eur. J. Oper. Res., 96, 205–216, 1997.

[20] Błażewicz J., Lenstra J., Kan A.R., *Scheduling subject to resource constraints – classification and complexity*, Discret. Appl. Math., 5, 11–24, 1983.

[21] Hartmann S., Kolisch R., *Experimental evaluation of state-of-the-art heuristics for the resource-constrained project scheduling problem*, Eur. J. Oper. Res., 127, 394–407, 2000.

[22] Kolisch R., Hartmann S., *Experimental investigation of heuristics for resource-constrained project scheduling: an update*, Eur. J. Oper. Res., 74, 1, 23–37, 2006.

[23] Nawaz M., Enscore E., Ham I., *A heuristic algorithm for the m machine, n-job flow-shop sequencing problem*, OMEGA, 11, 91–95, 1983.

[24] Woo D.S., Yim H.S., *A heuristic algorithm for mean flowtime objective in flowshop scheduling*, Comput. Oper. Res., 25, 175–182, 1998.

[25] Klimek M., Łebkowski P., *Insertion algorithms with justification for solving the resource-constrained project scheduling*, Decis. Mak. Manuf Serv., 10, 1–2, 31–43, 2016.

[26] Kolisch R., *Serial and parallel resource-constrained project scheduling methods revisited: theory and computation*, Eur. J. Oper. Res., 90, 320–333, 1996.

[27] Selle T., Zimmermann J., *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.

[28] Vanhoucke M., *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, pp. 1–23, 2006.

[29] Valls V., Ballestin F., Quintanilla S., *Justification and RCPSP: a technique that pays*, Eur. J. Oper. Res., 165, 2, 375–386, 2005.

[30] Valls V., Ballestin F., Quintanilla S., *Justification technique generalisations*, [in:] Józefowska J., Węglarz J. [Eds.], *Perspectives in Modern Project Scheduling*, Springer, Berlin, pp. 205–223, 2006.