Applied Computer Science, vol. 13, no. 4, pp. 20–34
doi: 10.23743/acs-2017-26

Submitted: 2017-10-08
Revised: 2017-10-19
Accepted: 2017-12-01

resource-constrained multi-stage project scheduling,
discounted cash flows, milestones, priority algorithms

Marcin KLIMEK[*]

PRIORITY ALGORITHMS FOR THE PROBLEM OF FINANCIAL OPTIMISATION OF A MULTI-STAGE PROJECT

Abstract

The article presents the problem of the financial optimisation of a multi-stage project from the contractor's perspective, where customer's payments are analysed as a cash inflow (contractor's revenues) after completing contractual stages and contractor's expenses incurred for the activities executed. In order to solve the problem, priority algorithms are proposed: single-pass and multi-pass ones, using different priority rules and techniques for generating solutions dedicated to the investigated optimisation model. The article presents a comparison of the effectiveness of individual algorithms in the case of adequately prepared test problems.

1. INTRODUCTION

One of the most frequently discussed optimisation issues is the problem of project scheduling with limited availability resources, known as RCPSP (Resource-Constrained Project Scheduling Problem). Various types of resources, methods of executing activities, optimisation criteria, etc. are analysed for RCPSP. An overview of the research can be found in review publications (Hartmann & Briskorn, 2012; Józefowska & Węglarz, 2006). In practice one of the more important aspects of planning a project is the financial optimisation of the project, where in the course of scheduling work activities considerations are given to all cash flows associated with the project, which, in most research studies, are discounted, i.e. the value of their actual NPV (Net Present Value) is calculated

[*] State School of Higher Education, Department of Computer Science, Sidorska 95-97, Biala Podlaska, m.klimek@dydaktyka.pswbp.pl

at an assumed discount rate. Models of maximising discounted cash flows, referred to as RCPSP-DC (RCPSP with Discounted Cash flows), PPS (Payment Project Scheduling) (Mika *et al.*, 2005; Ulusoy *et al.*, 2001) are considered.

Payment project scheduling is analysed here from the contractor's point of view with maximisation of the discounted cash flows for a project in which the customer establishes, in conjunction with the contractor, the contractual work stages (milestones): deadlines for their completion and the amounts of payments. In the settlements between the customer and the contractor, a penalty system is applied, whereby penalties are imposed for missing the contractual deadline for the realisation of a project stage, to motivate the contractor to complete the project as fast as possible. Failure to meet contractual deadlines for the realisation of project stages leads to charging penalties reducing customers's stage payments to the contractor. In spite of the risk of penalties for untimely execution, the proposed settlement model is beneficial to the contractor since it enables them to obtain earlier payments from the customer for performing the activity, which they can spend on current business, such as completing new activities, purchasing necessary materials, salary payments etc.

The proposed original model of project financial optimisation with defined milestones can be useful in practice. Its application may lead to increasing the control over the process and its timely realisation. The proposed RCPSP model, apart from the author's works, has not been considered in other research studies. In this model staged cash flows are analysed for project scheduling with multiple ways of executing activities, referred to as MMRCPSP (Multi-Mode RCPSP) (He *et al.* 2009).

This paper analyses the problem of scheduling a project with limited availability of renewable resources and with one way of executing the activity (single-mode RCPSP), adhering to the criterion of maximising the sum of discounted cash flows: with customer's payments for the completed stages of the project, with contractual penalties for not meeting the deadlines and with contractor's expenses incurred for activity realisation. The aim of the paper is to analyse the effectiveness of the developed priority algorithms for the problem under consideration, in which different priority rules and dedicated techniques for generating a solution are applied, using a forward or backward scheduling strategy with stage shift procedures, or schedule justification. Computational experiments are conducted for the test instances from the PSPLIB (Project Scheduling Problem LIBrary) (Kolisch & Sprecher, 1997), with additionally defined contractual project stages and defined cash flows for financial settlements of works.

2. FORMULATION OF THE PROBLEM

The classic problem of scheduling a project with nonpreemptive activities (tasks) and with one way of executing them is analysed (single-mode RCPSP). The proposed optimisation criterion is the maximisation of the sum of discounted cash flows from the contractor's point of view with the expenditure related to activity realisation and with revenues obtained for executing the contractual stages of the project:

$$F = \sum_{i=1}^{N_A} \frac{CFA_i}{(1+\alpha)^{ST_i}} + \sum_{m=1}^{N_M} \frac{CFM_m}{(1+\alpha)^{MT_m}}, \tag{1}$$

with the following constraints:

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

$$\sum_{i \in J(t)} r_{ik} \le a_k, \qquad \forall t : t = 1, ST_{N_A+1}, \forall k : k = 1,..K, \tag{3}$$

and for defined contractual stage settlements:

$$MT_m = \max_{i \in MA_m}(FT_i), \qquad \forall m : m = 1,\dots,N_M \tag{4}$$

$$CFM_m = MP_m - MC_m \cdot \max(MT_m - MD_m, 0), \qquad \forall m : m = 1,\dots,N_M, \tag{5}$$

where: F – objective function, sum of discounted cash flows from the perspective of the project contractor,
i – activity index, $i = 1, ..., N_A$ (N_A – number of activities),
m – index of the project stage, $m = 1, ..., N_M$ (N_M – number of stages),
CFA_i – the contractor's expenses related to the execution of activity i,
α – discount rate,
ST_i – starting time of the activity i,
CFM_m – customer's payment for the execution of the m-th project stage (revenues from the contractor's perspective) set for the current schedule,
MT_m – completion date of the m-th project stage in the planned schedule,
d_i – duration of the activity i,
E – a set of arcs showing the sequential dependencies between activities of finish-start zero-lag precedence type in the project representation shown as a directed graph G (V, E), wherein V is a set of edges corresponding to activities,
$J(t)$ – a set of activities which are executed during the period [t–1, t],

r_{ik} – demand of the activity i for resources of the type $k = 1 \dots K$ (K – number of resource types),
a_k – availability of renewable resources of the type k (throughout the duration of the project), the number of the resources used at each moment t cannot exceed a_k,
MA_m – a set of activities to be executed at the m-th project stage,
FT_i – completion date of the activity i ($FT_i = ST_i + d_i$),
MP_m – customer's payment to the contractor for completing m-th project stage,
MD_m – contractual deadline of the m-th project stage,
MC_m – contractual unit penalty for exceeding the deadline of the m-th stage of project MD_m.

The objective of scheduling is to maximise the sum of discounted cash flows (see: formula 1) with precedence constraints (see: formula 2) and resource constraints (see formula 3) and considerations for financial settlements: the contractor's expenses for the activities performed and the customer's stage payments.

In order to determine stage settlements, the following are determined (e.g. by negotiation between the customer and the contractor, whose aim is to find a solution that satisfies both parties), the MA_m group of activities to be performed at a given stage of the project, deadlines for their realisation MD_m, the amount of customer's payments MP_m and contractual unit penalties MC_m for delays in activity completion by the contractor. The system of stage settlements can be useful in practice, because of the benefits of its implementation, both for the customer and the contractor, it may lead to reducing the problem of untimely project completion.

To illustrate the analysed optimisation model, let us consider an example project consisting of 8 activities being realised with the use one resource type with availability equal to 10. This project is shown in fig. 1. using AON (Activity-On-Node) representation. Three contractual project stages (milestones) have been defined:

– stage 1, where $MA_1 = \{0, 1, 2\}$ activities are executed, with contractual completion date $MD_1 = 4$, for which the customer's payment is equal to $MP_1 = 50$, which may be reduced by the cost of possible work delays, calculated on the basis of unit cost $MC_1 = 4$,
– stage 2, at which $MA_2 = \{3, 4, 5\}$ activities are executed, with contractual completion deadline $MD_2 = 8$, for whose completion the customer pays the amount $MP_2 = 50$, which may be reduced by the cost of possible work delays, calculated on the basis of unit cost $MC_2 = 4$,
– stage 3, completing the project, upon which activities $MA_3 = \{6, 7, 8, 9\}$ are performed, within the contractual deadline $MD_3 = 12$, for which the customer makes the payment of $MP_3 = 80$, reduced by the cost of possible delays in project completion, calculated on the basis of unit cost $MC_3 = 10$.

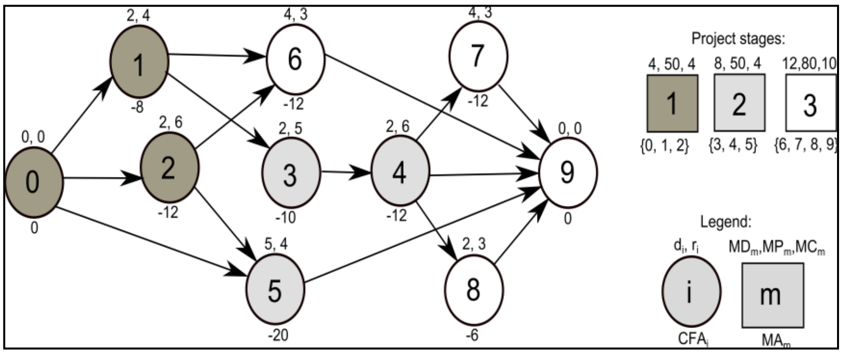

Fig. 1. An example project with staged settlements in AON representation

A discount rate α = 0.01 is adopted in the calculation of discounted cash flows.

In relation to the defined contractual cash flows connected with the project, its contractor develops a schedule in which, from their perspective, the sum of discounted cash flows *F* is maximised (see: formula 1). The growth of *F* leads to delaying expenses (negative cash flows) borne at the beginning of the activity (lower discounted value) and early acquisition of the customer's payments (positive cash flows) made upon the completion of project stages (higher discounted value). The following part of the article proposes techniques of generating schedules dedicated to the analysed financial problem of multi-stage project optimisation.

3. SCHEDULE GENERATION TECHNIQUES

The solution to the RCPSP (in direct representation) is usually a vector of activity starting time values, on the basis of which an objective function is determined, e.g. the duration of the project or the sum of the discounted cash flows of the project. When searching for a solution, a "convenient" indirect representation is applied, where schedules are encoded, most often as an activity list, a permutation of activities numbers with considerations for precedence relations. For a given activity list, a sequence that meets precedence and resource constraints is determined using decoding procedures SGS (Schedule Generation Scheme) i.e. serial SGS, parallel SGS etc. (Kolisch, 1996a). When building a solution with SGS, it is possible to use forward scheduling or backward scheduling. The analysis of algorithm efficiency and solution generation techniques for RCPSP can be found in review studies (Hartmann & Kolisch, 2000; Kolisch & Hartmann, 2006).

For RCPSP-DC, as well as for the analysed problem of the financial optimisation of a multi-stage project, forward or backward scheduling found through SGS procedures can be improved. Different techniques of generating solutions are used, i.e. bidirectional SGS, right shift or left shift algorithms, whose overview can be found in work (Vanhoucke, 2006). These techniques are used for developing schedules in which activities (stages) with assigned positive cash flows are started as soon as possible, while the activities (stages) related to the negative cash flows are planned for realisation as late as possible.

With regard to the analysed problem, from the contractor's perspective, it is advisable to collect the customer's payments for completed contractual work stages as soon as possible and incurring expenses related to commenced activities as late as possible (an increase of *F* always brings about postponing activities, which does not alter the time of completing project stages). Completing of project stages earlier than contractual deadlines stipulate may generate benefits owing to the higher NPV value of the customer's payments made earlier.

Due to the lack of procedures that would generate a solution suitable for the analysed model, the author develops and tests various techniques of creating schedules in their research: dedicated SGS procedures, algorithms activity shifting: right shift (forward scheduling) or left shift (backward scheduling) (Klimek & Łebkowski, 2015b), justification of schedules. In this work, two effective techniques are applied in the developed priority algorithms:

– justification which takes into consideration deadlines for completing contractual project stages,
– backward scheduling in the course of optimising (moving) completion times of contractual work stages (Klimek & Łebkowski, 2015a).

Justification techniques (Valls *et al.*, 2005) are used for improving the schedule, found using SGS, and are used for RCPSP, among others for the problem of minimising project duration or the problem with defined activity completion deadlines (RCPSP with Due Dates). One distinguishes RJ (Right Justification) and LJ (Left Justification). Justification of a given activity to the right (left) consists in determining the latest (earliest) possible starting time for this activity, with considerations for precedence and resource constraints. In this study activeities of maximum finish time (minimum starting time) are successively subjected to right (left) justification in a changing schedule (justification by extremes).

When generating solutions for RCPSP, techniques such as double justifycation by extremes, successively RJ+LJ or LJ+RJ are used. For the analysed problem it is necessary to modify RJ - when the activities are justified, the current work stage completion time should be considered. For a forward generated schedule, it is advisable to use triple RJ+LJ+RJ justification, which will transform the solution so that the activities are started as late as possible at the earliest possible completion of contractual project stages.

The effect of triple RJ+LJ+RJ justification is shown in fig. 2d.

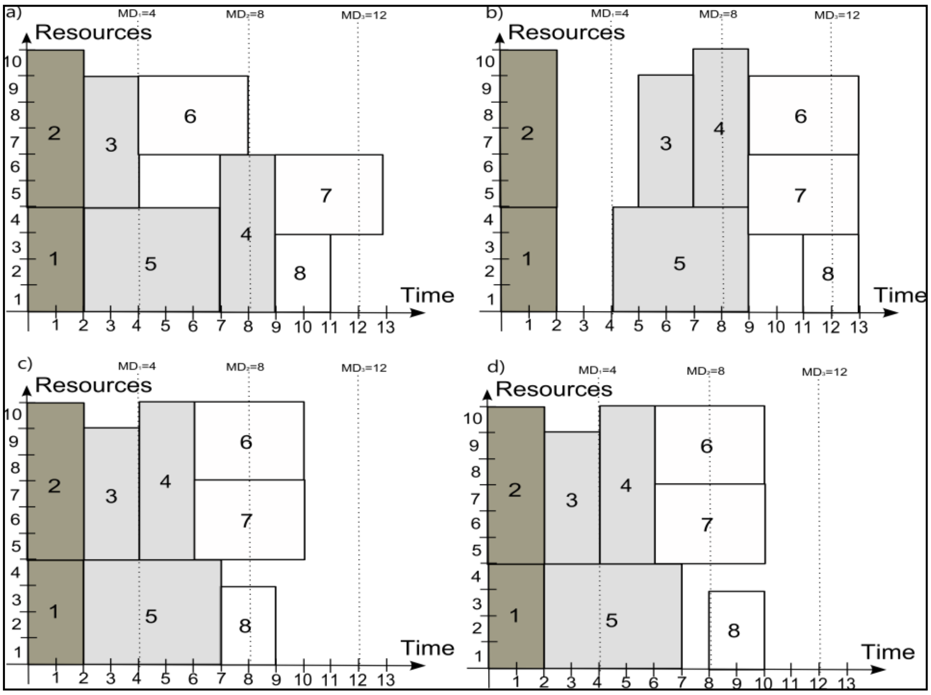

Fig. 2. a) The forward schedule generated using the serial SGS for the activity list {1, 3, 5, 2, 6, 4, 7, 8}; b) Schedule after RJ; c) Schedule after RJ+LJ; d) Schedule after RJ+LJ+RJ

In fig. 2a a schedule generated using the SGS serial procedure for the activity list {1, 3, 5, 2, 6, 4, 7, 8} (value of the objective function for this solution $F = 63.94$) is presented. This schedule is subjected to right justification, for which successive activities with a maximum time of completion are chosen – activities 7, 8, 4, 6, 5, 3, 2, 1 (when they have the same completion time, the activity with a higher number is analysed earlier). The modified technique of justification to the right of a given activity consists in setting the starting time as late as possible, so as not to exceed the current completion time of the project stage ($MT_1 = 2$, $MT_2 = 9$, $MT_3 = 13$), to which this activity belongs, taking into account the precedence and resource constraints. As a result of right justification, thanks to the postponed start of activities 3, 5, 6 and 8, a schedule with a higher value of the objective function $F = 65.29$ is created.

Then left justification is performed for the schedule from fig. 2b which is followed by successive activities with a minimum starting time of respectively 1, 2, 5, 3, 4, 6, 7, 8 (where the starting time is equal, the activity with a lower number is analysed first). As a result of LJ, the schedule shown in fig. 2c with a higher value of the objective function $F = 78.88$ is created due to the earlier execution of stages 1, 2 and 3 ($MT_1 = 2$, $MT_2 = 7$, $MT_3 = 10$) and meeting contractual deadlines for their completion.

The schedule from fig. 2c is subjected to a modified RJ technique with the assumption of new completion times of contractual project stages (MT_1 = 2, MT_2 = 7, MT_3 = 10). Activities 7, 6, 8, 5, 4, 3, 2, 1 undergo successive justification, and the new schedule is created, shown in fig. 2d, with the objective function value F = 78.94, higher than the schedule in fig. 2c, due to the later start of the activity 8. The schedule in fig. 2d is the final solution to the analysed problem obtained by using the triple justification technique for the activity list {1, 3, 5, 2, 6, 4, 7, 8}.

The backward scheduling procedure for optimising (shifting) the completion times of contractual work stages is yet another technique for generating solutions used in this paper. This technique performs unitary time shifts of the planned completion time MT_m. Individual shifts of all the stages are analysed starting from the first one and ending with the last one. They continue to be conducted as long as this operation increases the value of the function F. The process of optimising completion times of contractual work stages is illustrated in fig. 3.

The backward schedule arranged for the analysed activity list {1, 3, 5, 2, 6, 4, 7, 8}, the serial SGS, with such completion times for stage completion as contractual deadlines, are shown in fig. 3a (value of the objective function for this solution F = 77.52). The procedure of the left shifts for the stages of the project is as follows:

- the left shift of the first stage, assuming the shift of the MT_1 = 3 for stage completion by one unit, the objective function value F increases from 77.52 to 77.92, assuming the shift of the MT_1 = 2 stage completion time by a successive unit, the objective function value F increases from 77.92 to 78.22 due to the increased discounted first stage payment, it is not possible to complete the first stage within the MT_1 = 1 due to the duration of activities executed at this stage (d_1 = 2, d_2 = 2), the procedure proceeds to the second stage of works;
- the left shift of the second stage, assuming a shift of the MT_2 = 7 by one unit increases the value of the objective function F from 78.22 to 78.28 due to the increased discounted second stage payment, it is not possible to complete the second stage within the MT_2 = 6 because with such setup the starting time of activity 1 or activity 2 is determined as "negative", the procedure proceeds to the third stage of works;
- the left shift of the third stage, assuming a shift of the MT_3 = 11 stage completion time by one unit, the objective function value F increases from 78.28 to 78.71, assuming a shift of the MT_3 = 10 stage completion time by a successive unit, the objective function value F increases from 78.71 to 78.94 due to the increased discounted payment for the third stage, it is not possible to complete stage within the MT_3 = 9 because at this time the starting time of activity 1 or activity 2 determined by the backward procedure is "negative", the algorithm terminates.

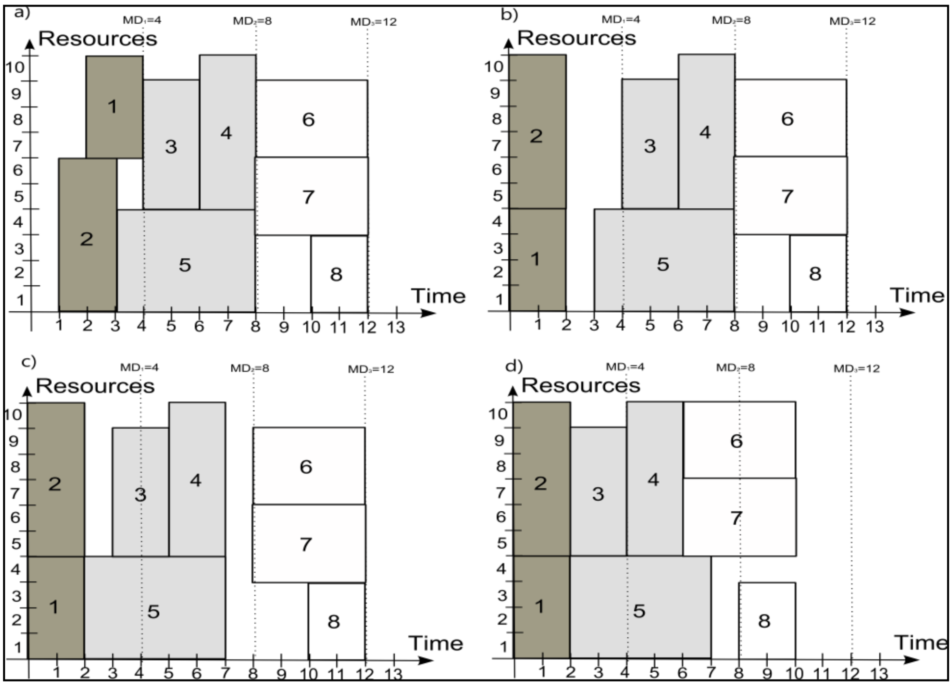

Fig. 3. a) The backward schedule generated using the serial SGS for the activity list {1, 3, 5, 2, 6, 4, 7, 8} and at the completion times for stages $MT_1 = MD_1 = 4$, $MT_2 = MD_2 = 8$, $MT_3 = MD_3 = 12$; b) The backward schedule determined as a result of the optimisation of the completion time for the first stage of the project $MT_1 = 2$, $MT_2 = 8$, $MT_3 = 12$; c) The backward schedule determined as a result of the optimisation of completion times for the first and the second stage of the project for $MT_1 = 2$, $MT_2 = 7$, $MT_3 = 12$; d) The backward schedule determined as a result of the optimisation of completion times for all the stages of the project for $MT_1 = 2$, $MT_2 = 7$, $MT_3 = 10$.

The schedule presented in fig. 3d was created after subjecting the project stages to left shifts in time.

4. PRIORITY ALGORITHMS

The problem of scheduling a project with limited resources as a generalisation of the job shop problem is a considerably NP-Hard problem (Błażewicz et al., 1983), for which it is more practical to use effective approximate heuristic algorithms finding solutions within an acceptable time period, especially for larger problems. A review of the heuristics used for RCPSP and a comparison of their effectiveness can be found in literature (Hartmann & Kolisch, 2000; Kolisch & Hartmann, 2006).

Amongst the heuristic algorithms one may distinguish constructive heuristics (priority, insertion) which create inaugural solutions for local search algorithms (simulated annealing, genetic algorithms). The aim of this work is to establish the effective priority heuristics that use SGS procedures and presented scheduling techniques that construct a schedule based on an activity list built in an ascending order of priorities of particular activities calculated for the adopted priority rules. The following procedures are analysed:

– single-pass ones - creating one solution based on an activity list arranged in an ascending order of activity priorities, they are quick and very easy to implement but they are characterised by low efficiency, even for the most effective priority rules: LFT (Latest Finish Time), EFT (Earliest Finish Time), MTS (Most Total Successors) and MTSPT (Most Total Successors Processing Time),
– multi-pass, X-pass ones - creating more potential schedules, algorithms that apply multiple priority rule methods at the same time, or sampling methods in which priority activities are randomly assigned with considerations for the applied priority rule.

One pass of the proposed multi-pass heuristics, in which one activity list *L* is generated, may be described as follows (Klimek, 2010):

Step 1:
Setting an empty activity list *L*. Placement of all of the successors with the initial operation number 0 on the list *LA* in the order determined by the priorities of the activities determined for the applied priority rule.

Step 2:
Selecting randomly one activity *i* from the list *LA* with considerations for "chances" determined on the basis of the ranking of the activities on the list *LA*: the first activity has the highest number of chances for selection, equal to the number of activities on the list *LA*, ..., the last activity has one chance for selection.

Step 3:
Deletion of the randomly selected activity *i* from the list *LA* and putting it on the list *L*. Adding all the successors of the activity *i*, of which all the predecessors are already on the list *L*, to the *LA* retaining the order established on the basis of the priorities of the activities assigned to the applied priority rule.

Step 4:
Repetitions of steps 2-3 until the list *L* is complete with all the activities carried out in the project.

In each pass, the resultant activity list L is decoded using SGS procedures and modified using solution generating techniques dedicated to the analysed problem (forward scheduling and justification technique, backward scheduling with optimisation of stage completion times). The result of a multi-pass algorithm is the schedule of the highest value of the objective function F (see formula 1), determined for all the passes from the analysed activity lists L.

The effectiveness of the developed heuristics can be influenced by the applied priority rules (Kolisch, 1996b). Table 1 presents the priority rules (rules R_0–R_6), developed for the investigated problem of the financial optimisation of a multi-stage project, which are used in computational experiments.

Tab. 1. Priority rules

Rule	Rule description
R_0	random priorities of activities
R_1	the minimum latest starting time of the activity with considerations for the contractual deadlines for the completion of project stages
R_2	minimum latest deadline for the completion of the activity with considerations for the contractual deadlines for the completion of project stages
R_3	maximum number of all the successors of the activities
R_4	the maximum sum of the duration periods of a given activity and all of its successors
R_5	the minimum number of the project stage in which the activity is performed, the minimum cost of the activity CFA_i at the same stages
R_6	minimum cost of the performance of the activity

If, when using a given priority rule, the activities have the same priorities, the activities marked with lower numbers are listed in the earlier position in the activity list.

5. COMPUTATIONAL EXPERIMENTS

The experiments were conducted with a computer Intel Core I7, 3.0 GHz, 8 GB RAM and using an application implemented in the C# programming language of the Visual Studio.NET environment for test instances from the PSPLIB within the set J30 (480 30-activity instances) and J90 (480 90-activity instances). The manner of defining contractual project stages for each project from PSPLIB is the following:

- baseline schedule S is created using the serial SGS procedure for an activity list sorted in the ascending order of activity numbers {1, 2, ..., 30} for the set J30 or {1, 2, ..., 90} for the set J90,
- three stages of project are set for the accepted deadlines: $MD_1 = T/3$, $MD_2 = 2T/3$ and $MD_3 = T$, where T is the duration of the project in the baseline schedule S, successive sets of activities MA_m performed at individual

stages are determined on the basis of the baseline schedule S. The set MA_1 includes all the activities whose completion time is lower than or equal to MD_1. The set MA_2 includes all the activities whose completion time is lower than or equal to MD_2 and greater than MD_1, while 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, while the costs related to the execution of activitiess CFA_i are determined as proportional to the total demand for resources and time spent on a given activity assuming their sum for all the activities is 100:

$$CFA_i = \frac{d_i \cdot \sum_{k=1}^{K} d_{ik}}{\sum_{j=1}^{N_A} (d_j \cdot \sum_{k=1}^{K} r_{jk})} \cdot 100, \quad \text{for} \quad i = 1 \ldots N_A . \tag{6}$$

The discount rate assumed in the experiments is: $\alpha = 0.01$.

The purpose of the experiments is to verify the effectiveness of the proposed priority heuristics (single-pass and multi-pass), to find the best priority rules (from amongst the rules R_0–R_6) and techniques of generating solutions dedicated to the investigated problem. In randomised multi-pass heuristics for each instance of the problem, the number of verified solutions (passes) is equal 500. In table 3 and 5 are presented average values of the objective function F for 960 experiments (two schedules are generated for each 480 test instances). The results of the computational experiments are presented in tables 2–5.

Tab. 2. The results of the computational experiments for single-pass priority algorithm for the projects in the set J30

	Forward, serial SGS		Forward, parallel SGS		Backward, serial SGS	
	RJ	RJ+LJ+RJ	RJ	RJ+LJ+RJ	–	LJ+RJ
Rule R_0	54.73	66.75	63.26	69.47	62.24	71.63
Rule R_1	67.02	71.41	69.31	72.26	62.81	73.51
Rule R_2	69.57	73.05	69.72	72.51	64.41	74.22
Rule R_3	65.67	71.49	68.79	71.71	63.31	73.29
Rule R_4	63.44	70.27	67.88	71.59	63.64	73.15
Rule R_5	64.11	69.83	67.63	71.19	61.34	72.85
Rule R_6	56.34	66.73	64.20	69.72	62.86	72.08

Tab. 3. The results of the computational experiments for multi-pass priority algorithm for the set J30

	Forward, serial SGS		Forward, parallel SGS		Backward, serial SGS	
	RJ	RJ+LJ+RJ	RJ	RJ+LJ+RJ	–	LJ+RJ
Rule R_0	75.71	77.13	75.29	76.67	77.08	77.45
Rule R_1	77.13	77.47	75.97	76.96	77.17	77.52
Rule R_2	77.13	77.48	75.97	76.99	77.28	77.55
Rule R_3	76.63	77.33	75.71	76.84	77.14	77.53
Rule R_4	76.56	77.26	75.65	76.81	77.09	77.52
Rule R_5	76.57	77.40	75.78	76.94	77.30	77.51
Rule R_6	75.69	77.26	75.38	76.80	77.24	77.48

Tab. 4. The results of the computational experiments for single-pass priority algorithm for the projects in the set J90

	Forward, serial SGS		Forward, parallel SGS		Backward, serial SGS	
	RJ	RJ+LJ+RJ	RJ	RJ+LJ+RJ	-	LJ+RJ
Rule R_0	19.78	39.85	32.27	42.51	29.19	47.95
Rule R_1	36.58	46.17	40.35	46.76	28.99	49.10
Rule R_2	37.78	46.79	41.08	47.16	29.19	49.55
Rule R_3	29.41	43.18	36.65	44.22	29.55	48.73
Rule R_4	27.43	42.90	35.51	43.85	29.25	48.47
Rule R_5	31.55	44.60	38.18	45.54	28.93	48.53
Rule R_6	21.14	40.16	32.55	42.68	29.01	47.36

Tab. 5. The results of the computational experiments for multi-pass priority algorithm for the set J90

	Forward, serial SGS		Forward, parallel SGS		Backward, serial SGS	
	RJ	RJ+LJ+RJ	RJ	RJ+LJ+RJ	–	LJ+RJ
Rule R_0	42.80	50.87	46.17	50.83	50.58	52.49
Rule R_1	49.71	52.25	49.66	51.96	50.71	52.63
Rule R_2	49.37	52.25	49.53	52.00	51.15	52.64
Rule R_3	45.84	51.07	47.66	51.11	50.22	52.58
Rule R_4	45.98	50.78	47.52	50.89	50.00	52.60
Rule R_5	46.73	52.30	48.52	51.97	51.63	52.59
Rule R_6	41.80	51.45	46.30	51.00	51.40	52.53

Applying a multi-pass priority algorithm significantly improves the quality of the generated schedules. The best results are achieved for multi-pass heuristics using combined solution generation techniques: backward scheduling with optimisation of contractual activity completion times and then improvement of the solution by double justification LJ+RJ. For the priority rule R_2 (minimum latest deadline for the completion of the activity), which turned out to be the most effective for both the set J30 and J90, in two experimental studies, this algorithm found 352 or 362 (108 or 102) best solutions from amongst those generated by all analysed heuristics for 480 test instances investigated for the set J30 (J90).

Using effective priority rules improves the quality of the obtained solutions. Schedules generated for algorithms using a rule with random priority activities R_0 are characterised by poor quality – the lowest average value of the objective function *F*. The effective priority rules are R_1, R_2, R_5. Quality improvement of the solutions is always observed after applying triple justification of schedules.

Using the schedules found by the proposed priority algorithms as initial solutions for metaheuristics can bring good results. It will be the subject of further research studies by the author.

6. CONCLUSIONS

In this paper, the problem of discounted cash flows maximising for a multi-stage project has been discussed from the contractor's perspective. The proposed optimisation model takes into account the expenses assigned to activities and client's payments for the completed stages of the project.

Single-pass and multi-pass priority algorithms for the analysed problem have been presented and tested. These algorithms use techniques of generating schedules with considerations for the specific nature of the discussed problem: activities should be planned as late as possible but with the earliest completion times of the project stages. Appropriate justification and backward scheduling techniques, which take into consideration completion times of contractual project stages, have been developed. The efficiency of priority algorithms and procedures of generation solutions has been verified for test instances from the PSPLIB library. Numerical experiments have confirmed good efficiency of the backward scheduling procedure with optimisation of contractual stages completion times and then improvement of the solution by double justification LJ+RJ.

The investigated issues are important and the results of the research work may be useful in the execution of practical projects in which stage settlements are beneficial for both the contractor and the customer. The proposed priority heuristics will be used to create inaugural solutions in metaheuristics, i.e. simulated annealing, which will be the subject of further research by the author.

REFERENCES

Błażewicz, J., Lenstra, J., & Kan, A. (1983). Scheduling subject to resource constraints classification and complexity. *Discrete Applied Mathematics*, 5, 11–24.

Hartmann, S., & Briskorn, D. (2012). A Survey of Variants and Extensions of the Resource-Constrained Project Scheduling Problem. *European Journal of Operational Research*, 207(1), 1–14.

Hartmann, S., & Kolisch, R. (2000). Experimental evaluation of state-of-the-art heuristics for the resource-constrained project scheduling problem. *European Journal of Operational Research*, 127, 394–407.

He, Z., Wang, N., Jia, T., & Xu, Y. (2009). Simulated annealing and tabu search for multimode project payment scheduling. *European Journal of Operational Research*, 198(3), 688–696.
Józefowska, J., & Węglarz, J. (eds.) (2006). Perspectives in modern project scheduling. Springer.
Klimek, M. (2010). Predyktywno-reaktywne harmonogramowanie produkcji z ograniczoną dostępnością zasobów, PhD disertation, AGH, Cracow. (in Polish)
Klimek, M., & Łebkowski, P. (2015a) Harmonogramowanie projektu rozliczanego etapowo. AGH, Cracow. (in Polish)
Klimek, M., & Łebkowski, P. (2015b). Heuristics for project scheduling with discounted cash flows optimisation. *Bulletin of the Polish Academy of Sciences Technical Sciences*, 63(3), 613–622.
Kolisch, R. (1996a). Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation. *European Journal of Operational Research*, 90, 320–333.
Kolisch, R. (1996b). Efficient priority rules for the resource-constrained project scheduling problem. *Journal of Operations Management*, 14, 179–192.
Kolisch, R., & Hartmann, S. (2006). Experimental Investigation of Heuristics for Resource-Constrained Project Scheduling: An Update. *European Journal of Operational Research*, 74(1), 23–37.
Kolisch, R., & Sprecher, A. (1997). PSPLIB – a project scheduling library. *European Journal of Operational Research*, 96, 205–216.
Mika, M., Waligóra, G., & Węglarz, J. (2005). Simulated annealing and tabu search for multi-mode resource-constrained project scheduling with positive discounted cash flows and different payment models. *European Journal of Operational Research*, 164(3), 639–668.
Ulusoy, G., Sivrikaya-Serifoglu, F., & Sahin, S. (2001). Four Payment Models for the Multi-Mode Resource Constrained Project Scheduling Problem with Discounted Cash Flows. *Annals of Operations Research*, 102, 237–261.
Valls, V, Ballestin, F, & Quintanilla, S. (2005). Justification and RCPSP: a technique that pays. *European Journal of Operational Research*, 165(2), 375–386.
Vanhoucke, M. (2006). A scatter search procedure for maximizing the net present value of a resource-constrained project with fixed activity cash flows. *Working Paper 2006/417*, Gent, 1–23.