Article

Financial Optimization of the Resource-Constrained Project Scheduling Problem with Milestones Payments

Marcin Klimek

Department of Computer Science, State School of Higher Education in Biala Podlaska,
21-500 Biala Podlaska, Poland; m.klimek@dydaktyka.pswbp.pl

Featured Application: The proposed model of stage settlements with milestones payments may be useful in practical projects and be beneficial both for the client and for the contractor. Its application may reduce delays in the completion of project stages, which is a significant problem in information technology (IT) or industry projects.

Abstract: This article presents the resource-constrained project scheduling problem with the discounted cash flow maximization criterion from the perspective of a contractor. Cash flows are considered as the contractor's expenses related to the execution of activities and client's payments (revenue to the contractor) after the completion of contractual stages. To solve the problem, dedicated techniques to generate solutions and a simulated annealing algorithm are proposed. Finally, the proposed procedures are examined using the test library, Project Scheduling Library (PSPLIB). An experimental analysis identified the efficient moves and techniques for creating solutions, that is backward scheduling with optimization of completion times of project stages and triple justification.

Keywords: resource-constrained project scheduling problem; discounted cash flow maximization; milestones payments; simulated annealing algorithm

Citation: Klimek, M. Financial Optimization of the Resource-Constrained Project Scheduling Problem with Milestones Payments. *Appl. Sci.* **2021**, *11*, 661. <https://doi.org/10.3390/app11020661>

Received: 18 December 2020

Accepted: 11 January 2021

Published: 12 January 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The resource-constrained project scheduling problem (RCPSP) is one of the most frequently undertaken issues within operational research. For RCPSP, optimizing models are often analyzed which only include time criteria, that is makespan minimization etc. [1]. In practice, however, an analysis of the financial effects of planned projects, in particular large-scale projects executed over a long period of time, plays an important role. For long-term projects it is recommended to take into account the change in the value of money over time, which occurs in Max-NPV (net present value maximization) models [2]. The optimization criterion of NPV maximization is the most common criterion that takes into account the financial aspects of project planning [3]. When calculating NPV, the sum of all discounted cash flows associated with the project is calculated. RCPSP models with discounted cash flows (RCPSP-DC) are considered. An overview of NPV maximization research can be found in the literature [4,5].

Research works analyze payment project scheduling (PPS) models from the perspective of the contractor and/or the client. Payment schedules determine the frequency, amounts and delivery dates of client payments to the contractor. There are different methods for making financial settlements by compromise between the client and the contractor, whose interests in respect of the amount, timing and location of progress payments are divergent. The contractor aims to obtain the highest possible client payments as soon as possible. The client prefers to make a one-off payment after work. According to Szmerekovsky [6] both parties (client and contractor) are willing to maximize their profit instead of looking for a plan that satisfies them both. PPS problems are analyzed, where the solution from the perspective of client [7], the contractor [4,8,9] or both is sought [10–13].

This work discusses the problem of financial optimization from the contractor's perspective. Contractor expenditures (cash outflows) are generally known or may be estimated by them prior to the project, while client payments are determined by negotiation. Different approaches are considered in which total costs for executing activities may be incurred at their beginning [6,9] or completion [14], or may be proportionally allocated to the beginning and end of their execution [15,16], or distributed depending on their progress [11]. Fixed costs related to the activities are most often incurred at the start time and other costs, such as resource consumption, may arise during their execution.

As contractors are forced to spend money on project activities, it is advisable for them to acquire as much money as possible from the client prior to completing the project, such as, funds corresponding to the progress of the project. An improperly designed project settlement plan between the contractor and the client can cause financial problems for the contractor. Payments can be made in many ways [6,9,14,17]. One-off lump-sum payments after completion of a lump sum payment (LSP) project are one of the forms of settlement between client and contractor, most preferred by the client. For long-term projects, the LSP model may be unacceptable to a contractor who has to spend a long time financing the activities. In practice, progress payments are often determined throughout the duration of the project. Earlier client payments are typically associated with certain events and with milestones (completion of specific activities, stages). In general, payments can be made according to a fixed schedule, such as once a month (Equal Time Intervals—ETI model, Progress Payments—PP model) or associated with specific events (Payments at Event Occurrences—PEO model), i.e., completion of activity (Payments at Activities' Completion times—PAC model) etc. [9,17,18].

PPS models serve as a decision support tool for negotiations between the client and the contractor and can be used when planning the contractor's budget. Elmaghraby [19] suggests that the contractor can use information about the project and its costs to construct an appropriate client-contractor payment model where the delivery dates for these payments are assigned to the various milestones and the amounts are estimated based on the cost of performance of activities. All progress payments are arranged during the scheduling at milestones specified while planning or payment models are created that allow contractors to identify milestones and related cash flows.

For the project settlement between the client and the contractor, the bonus-penalty system [8,20,21] can be used to encourage the contractor to execute project (stages) as soon as possible. Penalties are determined for exceeding the contractual completion time of the project (stage), bonuses are given for early completion of the project (stage), etc. Contractual deadlines are specified for the project (stage) or time windows are defined where project (stage) completion is not awarded or penalized.

For the RCPSP also the capital constrained project scheduling (CCPSP) models are analyzed [22,23], where capital is one of the limited, non-renewable resources that is considered in the scheduling. At any time during the project, cash inflows and outflows must balance. Activities can be executed only if funds are available, such as those obtained by completing previous activities or stages of the project.

In this work we propose the RCPSP model with the criterion of maximizing the discounted cash flows associated with the project from the contractor's perspective, with stage settlements in which client agrees with the contractor the work stages (milestones) with deadlines and the amounts of payment for completion of stage work. In the event of late completion of the contractual stages, penalties reducing stage client payments are applied. The proposed optimization model can be useful in practice where milestones are an important part of work planning, for example, to be used to determine the progress of a project. In the RCPSP studies, the stage settlement of project work, apart from the author's work, was not considered in this form. Milestone payments are considered in studies but optimization models differ from those discussed in this work [16,24]. Milestone events are often assigned to events such as completion of individual activity [4,19]. In this work, payments are made after the completion of the activity group as a stage of the project [8,21,25,26].

For the proposed financial model used to optimize a multi-stage project, it is recommended to schedule the work so that activities are started as late as possible at the earliest possible completion of contractual stages. The aim of this work is to check the efficiency of the simulated annealing algorithm using the techniques of generating solutions, taking into account the specific stage settlements, that is: backward scheduling with optimization of project completion times, justification and right-shifting of activities for schedule with fixed resource allocation.

The remainder of this paper is organized as follows. Section 2 contains a formulation of the problem. Section 3 presents the definition of an exemplary project illustrating the analyzed problem. Section 4 provides a simulated annealing algorithm and techniques for generating solutions that are presented for the example project. Section 5 describes the experimental analysis performed for the Project Scheduling Library (PSPLIB) test instances [27] with defined additional project financial settlements. At the end of the work, a summary is presented and directions for further study are provided.

2. Problem Formulation

A project is a unique set of activities executed using available resources (employees, machines, materials). This work discusses the classic non-preemptive single-mode RCPSP. The project is shown as a directed graph $G(V, E)$, in representation of activity-on-node (AON), where V is a set of nodes corresponding to the activities and E is a set of arcs describing the order relations between the activities of finish-start zero-lag precedence. Activities are executed using the limited, renewable resources whose number is constant over time. The schedule (start times for all activities) is determined with the optimization criterion (makespan minimization, DCF maximization etc.) when the order constraints (1) and resource constraints (2) are fulfilled.

$$ST_i + d_i \leq ST_j, \quad \forall (i, j) \in E \quad (1)$$

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

where:

- i is the activity index, $i = 1, \dots, N_A$,
- N_A is the number of activities in a project,
- ST_i is the start time of the activity i ,
- d_i is the duration of activity i ,
- $J(t)$ is the set of activities executed in time interval $[t - 1, t]$,
- a_k is the number of available resources of type k ; at any time t , the number of resources used must not exceed the number of available resources a_k for $k = 1, \dots, K$,
- r_{ik} is the demand of activity i for the resource of type k ,
- K is the number of types of renewable resources.

This work offers an optimization model from the perspective of the contractor, which applies stage and financial settlements between the client and the contractor. Arrangements regarding joint settlements may be conducted for the adopted baseline schedule, which is the basis for defining inter alia, the number of contractual stages of the project, the dates of their completion, the activities to be taken at these stages and the method of payment determination. The proposed optimization model maximizing the discounted cash flow from the contractor's perspective with the costs incurred for the completion of the activities and the revenue obtained for the completed stages is described by the Equations (3)–(5):

$$F = \sum_{i=1}^{N_A} (CFA_i \cdot e^{-\alpha \cdot ST_i}) + \sum_{m=1}^{N_M} (CFM_m \cdot e^{-\alpha \cdot MT_m}) \quad (3)$$

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

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

where:

- F is the sum of discounted cash flows, objective function,
- CFA_i is the contractor's expenses related to the execution of activity i ,
- m is the index of project stage (milestone), $m = 1, \dots, N_M$ (N_M —number of stages),
- FT_i is the finish time of activity i ($FT_i = ST_i + d_i$),
- CFM_m is the client payment for completion of m —project stage established for the current schedule;
- α is the discount rate;
- MT_m is the completion time of m —project stage established for the current schedule;
- MA_m is the set of activities to be executed in m —project stage;
- MP_m is the agreed client payment to the contractor for completion of m —project stage;
- MD_m is the contractual deadline for m —project stage;
- MC_m is the contractual unit cost for exceeding MD_m deadline.

Client payments CFM_m (cash inflows from contractor's perspective) are made right after completion of contractual stages within MT_m completion time within defined amount MP_m reduced by any contractual penalties that include the delays unit cost of MC_m . It is assumed that all contractor's expenditures (cash outflows from contractor's perspective) are related to the execution of the activities and are incurred for each activity exactly as scheduled (in start time). The problem of possible delays in cash transfers is omitted in this model—for delayed payments, a system of non-periodic compensation may be used.

For contractors, the client payments for the project stages are financial means that can be used for the current business such as the purchase of materials necessary for the execution of subsequent activities, the salaries of the employees, etc. For the client, the need to incur earlier expenses is compensated for by the contractual penalties for failure to meet deadlines of the project stages and the possibility of controlling the course of work while the project is being executed.

3. Illustrative Example

To illustrate the analyzed optimization model we consider the sample project shown in Figure 1. The project consists of 8 activities and dummy activities 0 and 9 representing the starting and ending node in the graph $G(V, E)$ respectively. Activities are executed using one type of limited, renewable resources of availability of 10. Discount rate used in calculations is $\alpha = 0.01$.

Figure 1. Example of project in activity-on-node (AON) representation.

Assume that the baseline schedule, which is the basis for determining the stage settlement of the project between the client and the contractor, is shown in Figure 2. It can be generated, for example, using the schedule generation scheme (SGS) serial, for the activity list {1, 5, 2, 3, 6, 4, 7, 8}.

Figure 2. Baseline schedule.

Assume that based on the baseline schedule of Figure 2 three contractual stages of work that are distinguished that have the following completion times: $MD_1 = 4$, $MD_2 = 8$, $MD_3 = 12$. Activity sets to be completed in further stages are as follows $MA_1 = \{1, 2\}$, $MA_2 = \{3, 4, 5\}$, $MA_3 = \{6, 7, 8\}$. The cash flow related to CFA_i activities and the data for stage project settlements such as: contractual completion times MD_m , client contractual payment amounts MP_m and delay costs MC_m , are shown in the AON representation in Figure 3. Milestone data is denoted by the “square” symbol. The same colours show the nodes representing activities from the same stage of the project. The stage settlements should be arranged to encourage the contractor to execute the project as quickly as possible and not to exceed MD_m completion times.

Figure 3. Project in AON representation with defined stages.

Once the stage settlement has been completed, the contractor will prepare a schedule which maximizes the sum of the discounted cash flows function F . It is advisable to delay the expenditure (negative cash flow) incurred at the start of activities (lower discounted value) and accelerate client payments for project stages (positive cash flows) acquired at the time of their completion (higher discounted value). The baseline schedule shown in Figure 2 can be improved, for example, by the deferred start of activities 6 and 8, which will not delay the completion of stage 3. It is also possible to complete the work stages earlier. The proposed techniques of generating solutions taking into account the specificity of the discussed problem, are described further in this work.

4. Simulated Annealing Algorithm

The resource-constrained project scheduling problem as a job shop generalization is a strongly NP-hard problem [28]. Due to the complexity of the problem for RCPS, it is reasonable to use heuristic algorithms with polynomial computational complexity that find

solutions within acceptable time also for larger projects. A wide overview and comparison of the efficiency of algorithms for the problem of project scheduling can be found in [5,29].

In this work, when searching for schedules, a simulated annealing (SA) algorithm is used, which is effective and has repeatedly been used to solve the RCPSP problem [29]. The SA is flexible and has the ability (when it is fine-tuned) to approach global optimality; it explores the solutions space and prefers better neighbours but also accepts worse neighbours in order to avoid becoming stuck in local optima. The SA improves a single candidate's solutions. Population-based metaheuristic algorithms (scatter search—SS, genetic algorithm—GA [30], particle swarm optimization—PSO [31,32]) can be better for global searches due to global exploration and local exploitation ability and will be examined for proposed problems in future work.

The SA algorithm developed in this work is similar to the basic variant of this meta-heuristics [33]. The solutions are represented as a list of activities—permutation of activity numbers taking into account the precedence relations. The activity list is converted to an executable schedule using the SGS [34]. In this work, the serial SGS is used, which in the forward (backward) scheduling strategy, determines the starting time for the first (last) unscheduled activity from the activity list at the earliest (latest) time, taking into account the precedence and resource constraints.

The SA algorithm starts the search from the random list of activities for which the serial SGS generates a schedule. Then iterations of the algorithm are repeated until the stop condition is satisfied: viewing the specified maximum number of schedules. Each iteration of the SA algorithm performs a move generating a new activity list from the neighborhood of the current solution. Moves that will be used are:

- Insert—an activity is randomly selected, which is then moved to a random position between the last predecessor in the activity list and the first successor;
- Swap—an activity is randomly selected, which is then exchanged with a randomly selected activity taking into account precedence relations;
- Adjacent swap—a random activity is selected, which is then exchanged with the next activity in the activity list, unless it violates the precedence restrictions;
- Insert all—an activity is randomly selected, which is then moved to the position between the last predecessor in the activity list and the first successor; all possible insert positions are checked and the position for which the schedule with the highest value of objective function F is generated;
- Swap all—an activity is randomly selected, which is then exchanged with every activity occupying the last position between the last predecessor in the list and the first successor; this exchange operation is performed for which the schedule with the highest value of objective function F is generated.

For the problem of maximizing discounted negative and positive cash flows in the PAC model, the schedule established by the SGS procedures can be improved by starting activities (stages) related to the positive cash flow(s) as early as possible and by starting activities (stages) related to negative cash flow as late as possible. For this problem, what is analyzed are, among other things, bi-directional SGS [29,30], shift operation algorithms that move activities with negative cash flows to the right and/or activities with positive cash flows to the left [4], and recursive forward/backward improvement procedures [35,36].

For a schedule that is appropriate for the problem analyzed in the article, with objective function F , the positive cash flow, that is the client payment for completing the contractual stages of work, should be obtained as soon as possible and the negative cash flow, that is the contractor's expenses related to the activities started, incurred according to the as late as possible (ALAP) principle. When planning activities, it is advisable to carry out a "balance sheet" of losses from subsequent payments for the completed project stages and the benefits of deferred expenses for starting activities. Increase in NPV always defers these activities in time, the delayed start of which does not change the completion time of the project stages.

The development of effective procedures for generating schedules dedicated to the problem of financial multi-stage optimization is the subject of previous research [8,26,37]. The following subsections describe and illustrate the scheduling techniques used by the analyzed SA algorithm:

- backward scheduling with optimization of completion times of contractual project stages;
- forward scheduling with activity right-shift procedure;
- justification.

4.1. Backward Scheduling with Optimization of Completion Times of Stages

A baseline schedule is created for the processed activity list assuming scheduled completion times of MT_m equal to the contractual MD_m deadlines. In the baseline schedule, the MT_m stage completion times can be larger than the contractual MD_m , if it is not possible to generate the executable schedule for the processed activity list from $MT_m = MD_m$ for each $m = 1, \dots, N_M$. In the backward scheduling procedure with optimization of the completion times of the contractual stages of the project, the iterative, unit left-shift of planned MT_m are performed. Shifting operations are performed for the following stages, starting from the first and ending with the last. Unit shifts of a given stage are executed as long as this operation increases the value of the objective function F .

For an example project, the baseline schedule in Figure 2 is generated using the forward serial SGS. For backward serial SGS it is recommended to use a different activity list than for forward scheduling. It is suggested to set up a list of activities based on the baseline schedule—activities are sorted in the order of increasing activity finish times FT_i (similarly to the experimental studies for the initial solution). The activity list determined in this way is as follows: {1, 2, 5, 3, 4, 8, 6, 7}.

The backward schedule with the value of the objective function $F = 58.50$, determined for the activity list {1, 2, 5, 3, 4, 8, 6, 7} for completion times of stages such as: $MT_1 = MD_1 = 4$, $MT_2 = MD_2 = 8$, $MT_3 = MD_3 = 12$, is shown in Figure 4a. The procedure of left shifts of project stages starts from shifting the completion time of the first stage:

- for $MT_1 = 3$ a schedule with higher objective function is generated $F = 58.77$,
- for $MT_1 = 2$ a schedule with higher objective function is generated $F = 58.96$,
- for $MT_1 = 1$ it is not possible to create a feasible schedule and the procedure moves to the optimization of second stage and MT_2 completion time.

Figure 4. (a) Backward schedule for completion times of project stages $MT_1 = 4$, $MT_2 = 8$, $MT_3 = 12$, (b) schedule determined as a result of optimization of the completion time of first stage with $MT_1 = 2$, $MT_2 = 8$, $MT_3 = 12$.

The schedule established as a result of optimizing the completion time of the first stage with completion times of stages $MT_1 = 2$, $MT_2 = 8$, $MT_3 = 12$ is shown in Figure 4b. The optimization of the completion time of the second stage is as follows:

- for $MT_2 = 7$ the schedule with higher objective function $F = 58.98$ is generated;

- for $MT_2 = 6$ it is not possible to create a feasible schedule and the procedure moves to optimizing the completion time of the third stage MT_3 .

The schedule determined by optimizing the completion time of the first and second stage with $MT_1 = 2$, $MT_2 = 7$, $MT_3 = 12$ is shown in Figure 5a. The optimization of the completion time of the third stage is as follows:

- for $MT_3 = 11$ the schedule with higher objective function $F = 59.36$ is generated;
- for $MT_3 = 10$ the schedule with higher objective function $F = 59.59$ is generated;
- for $MT_3 = 9$ it is not possible to create a feasible schedule and the procedure ends the first step of the procedure for optimizing the completion times of project stages.

Figure 5. (a) Schedule determined as a result of optimization of the completion time of 2nd stage with $MT_1 = 2$, $MT_2 = 7$, $MT_3 = 12$, (b) schedule as a result of optimization of completion times of all stages: $MT_1 = 2$, $MT_2 = 7$, $MT_3 = 10$.

In the second step of the procedure, no changes are made (in optimizing of completion times of MT_1 , MT_2 , MT_3) to the schedule and the algorithm ends.

The final schedule generated during optimizing the completion times of all project stages with objective function $F = 59.59$ is shown in Figure 5b.

4.2. Forward Scheduling with Activity Right-Shift Procedure

Shifts of activities are performed taking into account the resource and order constraints. Shifting an activity to the right (or left) can cause various changes to the schedule. This problem does not occur if resource allocation is found for the schedule. Multiple resource allocations can be generated for a given schedule, characterized by various properties that affect the operations of right shifts and the quality of the resulting solutions. Assignment of resources to activities is analyzed during a proactive, robust scheduling [26,38–40]. An overview of resource allocation procedures is presented by Deblaere et al. [41]. For the analyzed problem, the resource allocation with as few additional order relations (additional arcs in $G(V,E)$) as possible are preferred as in robust scheduling. The new precedence restrictions resulting from the resource allocation reduce the possibility of shifting activities to the right to increase the cumulative cash flow F . To minimize the number of extra arcs, resource allocation algorithms allocate activities between which are order or unavoidable dependencies to be performed by the same resources, and maximize the sum of flows between the activities etc. [41]. For a schedule with a fixed resource allocation, it is easy to carry out the right-shift operations (resource constraints are taken into account)—unambiguously and with short calculation time which changes in the schedule will lead to starting activity shifted to the right can be determined.

In this work the resource allocation is generated using the ISH-UA (iterative sample heuristic with unavoidable arcs) procedure [7] whose effect on the schedule in Figure 2 is presented in Figure 6a.

Figure 6. (a) Schedule with resource allocation set by the ISH-UA (iterative sample heuristic with unavoidable arcs) procedure, (b) schedule set using the activity right-shift procedure.

The right-shift procedure is iterative. In subsequent iterations, right-shifting of activities considered in the sequence of decreasing time of their completion in the schedule for which the allocation of resources is determined is analyzed. For the analyzed activity, unit shifts are performed until the operation does not increase the value of F . For the resource allocation in Figure 6a, the right-shift of the subsequent activities is analyzed:

- activity 7 shift will not increase F ,
- one unit shift of activity 6 will increase F from 57.55 to 57.72,
- activity 8 shifts by two time units will increase F from 57.72 to 57.83,
- activity 4 and 3 shifts will not increase F ,
- one unit shift of activity 5 will increase F from 57.83 to 58.03,
- activity 2 and 1 shifts will not increase F .
- Schedule with right shifts with $F = 58.03$ is shown in Figure 6b.

The activity right-shift procedure does not make any changes to the completion of the project stages. It can be effective in improving schedules with the best possible MT_m . For the analyzed example, the optimal solution with $F = 59.59$ (Figure 7b) is found using the SGS serial for the activity list {1, 2, 3, 4, 5, 6, 7, 8} with the resource allocation found using the ISH-UA procedure (Figure 7a).

Figure 7. (a) Schedule generated for the activity list {1, 2, 3, 4, 5, 6, 7, 8} with the allocation of resources determined by the ISH-UA procedure. (b) Schedule improved by the right-shift procedure.

4.3. Justification

The right justification (RJ) and left justification (LJ) techniques transform the schedule generated by SGS procedures. Their use improves the quality of generated schedules [42]. They are used, among others, for RCPSP with a makespan minimization criterion or for

RCPSP with due dates. The justification of an activity to the right (to the left) consists in setting the latest (earliest) start time if possible taking into account the order and resource constraints. When generating solutions, RJ and LJ are combined: double justification is used, RJ + LJ or LJ + RJ respectively. Activities for RJ (LJ) can be selected:

- by any means from available activities whose all successors (predecessors) have already been subjected to justification when analyzing all activities from the beginning (end) of the schedule subjected to the justification (justification by eligible);
- in the order of decreasing finish times (increasing start times) in the schedule subjected to the justification (justification by extremes).

Most of the work, like this one, uses the effective justification by extremes technique, which is a special case of justification by eligibles.

Research conducted for the analyzed problem indicated that the triple justification technique (RJ + LJ + RJ) and the use of the modified RJ technique is effective for the forward schedule, and that the RJ technique is used to move an activity to the right (delaying its start), taking into account the current completion times of the project stages.

We will now show the RJ + LJ + RJ technique for the baseline schedule of Figure 2. First, RJ is used for activities with maximum finish time (justification by extremes)-activities in order 7, 6, 8, 4, 3, 5, 2, 1 (at the equal finish time, the activity with the higher number is analyzed earlier). For each activity, the latest start time is determined taking into account the precedence and resource relations and the actual completion time of the project stage to which this activity is assigned ($MT_1 = MD_1 = 4$, $MT_2 = MD_2 = 8$, $MT_3 = MD_3 = 12$). As a result of RJ, a schedule is presented in Figure 8a with the objective function $F = 58.50$.

Figure 8. (a) Baseline schedule after right justification (RJ), (b) baseline schedule after RJ + left justification (LJ), (c) baseline schedule after RJ + LJ + RJ.

Technique LJ is performed after RJ. The activities with the minimum start time for the schedule in Figure 8a (justification by extremes) are selected for LJ. Activities in the order 1, 2, 3, 5, 4, 6, 7, 8 are justified (at equal start time, the activity with lower number is analyzed first). For each activity, the earliest start time is determined, taking into account the precedence and resource relations. As a result of LJ, the schedule shown in Figure 8b with the higher value of the objective function $F = 59.53$ than the schedule of Figure 8a is generated, thanks to the earlier completion of stages 1, 2 and 3 ($MT_1 = 2$, $MT_2 = 7$, $MT_3 = 10$).

The schedule in Figure 8b can be improved by delaying the start of the activity while meeting the completion times of the contractual stages of the project. The activities are shifted by re-applying the modified RJ technique. For RJ, the activities are chosen with the maximum time of completion in the schedule from Figure 8b-activities in the order 7, 6, 8, 5, 4, 3, 2, 1. During the activity justification, current end times of project stages are considered ($MT_1 = 2$, $MT_2 = 7$, $MT_3 = 10$). As a result of RJ, the schedule presented in Figure 8c with a higher value of the objective function $F = 59.59$ than the schedule of Figure 8b is generated by the delayed start of activity 8.

5. Experimental Analysis

The experiments were conducted for 480 test instances from a J30 (30-activity projects) set and 480 test instances from J90 set (90-activity projects) from the PSPLIB [27]. For each PSPLIB project, three contractual stages are defined based on the baseline schedule S created using the SGS serial for the activity list $\{1, 2, \dots, 30\}$ or $\{1, 2, \dots, 90\}$. The contractual deadlines of stages are determined as $MD_1 = T/3$, $MD_2 = 2T/3$, $MD_3 = T$, where T is the makespan of the project in the baseline schedule S . The sets of activities to be executed in each stage are created as follows:

- MA_1 contains all activities whose end time in schedule S is less than or equal to MD_1 ,
- MA_2 contains all activities whose end time in schedule S is less than or equal to MD_2 and greater than MD_1 ,
- MA_3 includes other activities that are not included in MA_1 or MA_2 .

The data for determining cash flows for each test instance: contractual client payment amounts are $MP_1 = 60$, $MP_2 = 60$, $MP_3 = 120$, contractual penalties are $MC_1 = 1.5$, $MC_2 = 1.5$, $MC_3 = 3$, while the costs of executing CFA_i activity are calculated proportionally to the total demand for resources and time for a given activity, while the sum of CFA_i for all the activities is equal to 100. Discount rate used in experiments is $\alpha = 0.01$.

The experiments were conducted on an Intel Core i7-4770, 3.4 GHz CPU, 8 GB RAM using C# application implemented in Visual Studio 2017. The main goal was to evaluate the simulated annealing algorithm performance, moves (Swap, Insert, Swap all, Insert all, Adjacent swap), and techniques of generating schedules (backward scheduling, activity right-shift procedure, justification). Parameters for the SA algorithm such as: the geometric cooling scheme and the initial temperature of 0.1 were established in preliminary experiments.

In all analyzed configurations of the SA algorithm, 5000 activity lists are analyzed. Because of the stochastic nature of the calculations, experiments are conducted twice. The results of the computational experiments are presented in Tables 1 and 2. In these tables: FS—forward scheduling using serial SGS without improvement, BS—backward scheduling procedure with optimization of completion times of project stages, SH—forward scheduling with improvement using activity right-shift procedure, RJ—forward scheduling with improvement using right justification technique, TJ—forward scheduling with improvement using triple justification technique (RJ + LJ + RJ).

Table 1. Results of the experiments—J30 projects.

	Average Value of the Function F					Number of the Best Solutions					CPU
	Swap	Ins.	Swap All	Ins. All	Adj. Swap	Swap	Ins.	Swap All	Ins. All	Adj. Swap	Time [s]
FS, 1st run	76.19	76.13	76.17	76.02	75.91	0	0	0	0	0	0.05
FS, 2nd run	76.18	76.11	76.18	76.01	75.98	0	0	0	0	0	0.05
BS, 1st run	77.38	77.33	77.38	77.32	77.17	328	319	324	302	275	0.80
BS, 2nd run	77.38	77.32	77.37	77.30	77.16	329	306	322	299	270	0.81
SH, 1st run	77.25	77.16	77.21	77.08	76.99	55	48	51	46	39	4.46
SH, 2nd run	77.23	77.15	77.22	77.07	77.01	50	49	53	44	40	4.45
RJ, 1st run	77.50	77.43	77.49	77.36	77.29	275	255	271	234	223	0.17
RJ, 2nd run	77.48	77.44	77.46	77.37	77.29	266	255	263	243	218	0.17
TJ, 1st run	77.58	77.55	77.58	77.52	77.46	315	299	301	277	258	0.45
TJ, 2nd run	77.57	77.55	77.58	77.49	77.48	309	293	305	273	259	0.45

The average values of the objective function F for the baseline schedules are 74.04 for projects from J30 and 48.09 for projects from J90. The average values of the objective function F for the best found schedules are 80.63 for projects from J30 and 57.22 for projects from J90. The use of developed scheduling techniques improves the quality of generated solutions. The highest average value of the objective function F for both J30 and J90 project sets is reached for RS that uses the triple justification technique (RJ + LJ + RJ). An effective technique for generating solutions is also the backward scheduling procedure with the optimization of completion times of the project stages. It constructs solutions with a lower

mean value of F , but often generates schedules with the highest value of F . The least efficient and most costly in terms of calculation is the activity right-shift procedure.

Table 2. Results of the experiments—J90 projects.

	Average Value of the Function F					Number of the Best Solutions					CPU
	Swap	Ins.	Swap All	Ins. All	Adj. Swap	Swap	Ins.	Swap All	Ins. All	Adj. Swap	Time [s]
FS, 1st run	50.23	50.44	49.99	49.79	49.54	0	0	0	0	0	0.23
FS, 2nd run	50.23	50.44	50.03	49.82	49.57	0	0	0	0	0	0.23
BS, 1st run	52.88	52.96	52.85	52.67	52.53	133	209	89	65	21	4.44
BS, 2nd run	52.87	52.97	52.83	52.66	52.54	111	208	85	79	36	4.45
SH, 1st run	51.49	51.67	51.24	51.07	50.81	0	0	0	0	0	44.00
SH, 2nd run	51.46	51.67	51.27	51.07	50.85	0	0	0	0	0	44.01
RJ, 1st run	52.60	52.71	52.36	52.17	52.04	20	38	15	11	10	1.18
RJ, 2nd run	52.59	52.70	52.38	52.21	52.06	18	23	13	13	10	1.18
TJ, 1st run	52.94	53.03	52.82	52.78	52.63	54	82	27	25	10	3.21
TJ, 2nd run	52.95	53.00	52.86	52.81	52.65	59	75	37	32	13	3.22

For projects from J30 set, the best solutions are generated using Swap or Swap all moves, while for J90, the most effective is Insert move. The worst results are achieved for the Adjacent swap move.

6. Summary

This work examines the problem of maximizing the discounted cash flow for a multi-stage project from the perspective of its contractor with client payments made after the contractual stages of work, penalties for late completion and expenses incurred by the contractor when performing the activities. The problem is illustrated and techniques for generating appropriate solutions are described. A simulated algorithm based on dedicated scheduling methods is provided. An experimental analysis was performed, which identified the efficient moves (Swap, Swap all for J30, Insert for J90), and techniques for creating solutions, that is backward scheduling with optimization of completion times of project stages and triple justification.

The problem discussed is actual. The proposed model of project stage settlements can be useful in practice and beneficial for both the client and the contractor. It may lead to the reduced untimely execution of practical projects. The data used for financial settlements, such as deadlines, contractual payments and penalties can be negotiated between the client and the contractor to determine a solution that satisfies both parties.

Further works focus on comparing the effectiveness of various metaheuristics for the proposed model, especially population-based algorithms such as genetic algorithms and particle swarm optimization. Another direction of the work will be the development of the capital-constrained model (CCPSP) with staged project settlement for which the use of the described techniques of generating solutions will be tested.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hartmann, S.; Briskorn, D. A Survey of Variants and Extensions of the Resource-Constrained Project Scheduling Problem. *Eur. J. Oper. Res.* **2010**, *207*, 1–14. [\[CrossRef\]](#)
2. Russell, A.H. Cash Flows in Networks. *Manag. Sci.* **1970**, *16*, 357–373. [\[CrossRef\]](#)
3. Rosłon, J.; Książek-Nowak, M.; Nowak, P. Schedules Optimization with the Use of Value Engineering and NPV Maximization. *Sustainability* **2020**, *12*, 7454. [\[CrossRef\]](#)
4. Leyman, P.; Vanhoucke, M. A New Scheduling Technique for the Resource-Constrained Project Scheduling Problem with Discounted Cash Flows. *Int. J. Oper. Prod. Res.* **2015**, *53*, 2771–2786. [\[CrossRef\]](#)
5. Hartmann, S.; Kolisch, R. Experimental Evaluation of State-of-the-Art Heuristics for the Resource-Constrained Project Scheduling Problem. *Eur. J. Oper. Res.* **2000**, *127*, 394–407. [\[CrossRef\]](#)

6. Szmerekovsky, J.G. The Impact of Contractor Behavior on the Client's Payment-Scheduling Problem. *Manag. Sci.* **2005**, *51*, 629–640. [\[CrossRef\]](#)
7. Dayanand, N.; Padman, R. On Modelling Payments in Projects. *J. Oper. Res. Soc.* **1997**, *48*, 906–918. [\[CrossRef\]](#)
8. Klimek, M.; Łebkowski, P. Heuristics for Project Scheduling with Discounted Cash Flows Optimisation. *Bull. Pol. Acad. Tech.* **2015**, *63*, 613–622. [\[CrossRef\]](#)
9. Ulusoy, G.; Sivrikaya-Şerifoğlu, F.; Şahin, Ş. Four Payment Models for the Multi-Mode Resource Constrained Project Scheduling Problem with Discounted Cash Flows. *Ann. Oper. Res.* **2001**, *102*, 237–261. [\[CrossRef\]](#)
10. He, Z.; Wang, N.; Li, P. Simulated Annealing for Financing Cost Distribution Based Project Payment Scheduling from a Joint Perspective. *Ann. Oper. Res.* **2014**, *213*, 203–220. [\[CrossRef\]](#)
11. Kavlak, N.; Ulusoy, G.; Şerifoğlu, F.S.; Birbil, Ş.İ. Client-Contractor Bargaining on Net Present Value in Project Scheduling with Limited Resources. *Nav. Res. Log.* **2009**, *56*, 93–112. [\[CrossRef\]](#)
12. Ulusoy, G.; Cebelli, S. An Equitable Approach to the Payment Scheduling Problem in Project Management. *Eur. J. Oper. Res.* **2000**, *127*, 262–278. [\[CrossRef\]](#)
13. Bahrami, F.; Moslehi, G. Study of Payment Scheduling Problem to Achieve Client–Contractor Agreement. *J. Adv. Manuf. Technol.* **2013**, *64*, 497–511. [\[CrossRef\]](#)
14. Vanhoucke, M.; Demeulemeester, E.; Herroelen, W. Progress Payments in Project Scheduling Problems. *Eur. J. Oper. Res.* **2003**, *148*, 604–620. [\[CrossRef\]](#)
15. Dayanand, N.; Padman, R. Project Contracts and Payment Schedules: The Client's Problem. *Manag. Sci.* **2001**, *47*, 1654–1667. [\[CrossRef\]](#)
16. He, Z.; Wang, N.; Jia, T.; Xu, Y. Simulated Annealing and Tabu Search for Multi-Mode Project Payment Scheduling. *Eur. J. Oper. Res.* **2009**, *198*, 688–696. [\[CrossRef\]](#)
17. Leyman, P.; Vanhoucke, M. Payment Models and Net Present Value Optimization for Resource-Constrained Project Scheduling. *Comput. Ind. Eng.* **2016**, *91*, 139–153. [\[CrossRef\]](#)
18. 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.* **2005**, *164*, 639–668. [\[CrossRef\]](#)
19. Elmaghraby, S.E. Activity Nets: A Guided Tour through Some Recent Developments. *Eur. J. Oper. Res.* **1995**, *82*, 383–408. [\[CrossRef\]](#)
20. He, Z.; Xu, Y. Multi-Mode Project Payment Scheduling Problems with Bonus–Penalty Structure. *Eur. J. Oper. Res.* **2008**, *189*, 1191–1207. [\[CrossRef\]](#)
21. Klimek, M.; Łebkowski, P. Financial Optimisation of the Scheduling for the Multi-Stage Project. *Bull. Pol. Acad. Tech.* **2017**, *65*, 899–908. [\[CrossRef\]](#)
22. Smith-Daniels, D.E.; Padman, R.; Smith-Daniels, V.L. Heuristic Scheduling of Capital Constrained Projects. *J. Oper. Manag.* **1996**, *14*, 241–254. [\[CrossRef\]](#)
23. Leyman, P.; Vanhoucke, M. Capital- and Resource-Constrained Project Scheduling with Net Present Value Optimization. *Eur. J. Oper. Res.* **2017**, *256*, 757–776. [\[CrossRef\]](#)
24. He, Z.; Liu, R.; Jia, T. Metaheuristics for Multi-Mode Capital-Constrained Project Payment Scheduling. *Eur. J. Oper. Res.* **2012**, *223*, 605–613. [\[CrossRef\]](#)
25. Klimek, M. Insertion Algorithms to Solve the Resource-Constrained Multi-Stage Project Scheduling Problem with Discounted Cash Flow Maximization. *Manag. Prod. Eng. Rev.* **2018**, *9*, 109–116. [\[CrossRef\]](#)
26. Klimek, M.; Łebkowski, P. Robustness of Schedules for Project Scheduling Problem with Cash Flow Optimisation. *Bull. Pol. Acad. Tech.* **2013**, *61*, 1005–1015. [\[CrossRef\]](#)
27. Kolisch, R.; Sprecher, A. PSPLIB—A Project Scheduling Problem Library: OR Software—ORSEP Operations Research Software Exchange Program. *Eur. J. Oper. Res.* **1997**, *96*, 205–216. [\[CrossRef\]](#)
28. Blazewicz, J.; Lenstra, J.K.; Kan, A.H.G.R. Scheduling Subject to Resource Constraints: Classification and Complexity. *Discret. Appl. Math.* **1983**, *5*, 11–24. [\[CrossRef\]](#)
29. Kolisch, R.; Hartmann, S. Experimental Investigation of Heuristics for Resource-Constrained Project Scheduling: An Update. *Eur. J. Oper. Res.* **2006**, *174*, 23–37. [\[CrossRef\]](#)
30. Sari, V.; Cakir, S.; Kilic, E. Ece Evaluation of Scatter Search and Genetic Algorithm at Resource Constrained Project Scheduling Problems. In Proceedings of the 2011 15th IEEE International Conference on Intelligent Engineering Systems, Poprad, Slovakia, 23 June 2011; pp. 127–130.
31. Yu, Y.; Li, Y.; Li, J. Parameter Identification of a Novel Strain Stiffening Model for Magnetorheological Elastomer Base Isolator Utilizing Enhanced Particle Swarm Optimization. *J. Intell. Mater. Syst. Struct.* **2014**, *26*, 2446–2462. [\[CrossRef\]](#)
32. Chen, R.-M. Particle Swarm Optimization with Justification and Designed Mechanisms for Resource-Constrained Project Scheduling Problem. *Expert Syst. Appl.* **2011**, *38*, 7102–7111. [\[CrossRef\]](#)
33. Kirkpatrick, S.; Gelatt, C.D.; Vecchi, M.P. Optimization by Simulated Annealing. *Science* **1983**, *220*, 671–680. [\[CrossRef\]](#) [\[PubMed\]](#)
34. Kolisch, R. Serial and Parallel Resource-Constrained Project Scheduling Methods Revisited: Theory and Computation. *Eur. J. Oper. Res.* **1996**, *90*, 320–333. [\[CrossRef\]](#)
35. Vanhoucke, M.; Demeulemeester, E.; Herroelen, W. Maximizing the Net Present Value of a Project with Linear Time-Dependent Cash Flows. *Int. J. Oper. Prod. Res.* **2001**, *39*, 3159–3181. [\[CrossRef\]](#)
36. Vanhoucke, M. A Scatter Search Heuristic for Maximising the Net Present Value of a Resource-Constrained Project with Fixed Activity Cash Flows. *Int. J. Oper. Prod. Res.* **2010**, *48*, 1983–2001. [\[CrossRef\]](#)

-
37. Klimek, M. Techniques of Generating Schedules for the Problem of Financial Optimization of Multi-Stage Project. *Appl. Comp. Sci.* **2019**, *15*, 18–35. [[CrossRef](#)]
 38. Leus, R.; Herroelen, W. Stability and Resource Allocation in Project Planning. *IIE Trans.* **2004**, *36*, 667–682. [[CrossRef](#)]
 39. Klimek, M.; Łebkowski, P. Resource Allocation for Robust Project Scheduling. *Bull. Pol. Acad. Tech.* **2011**, *59*, 51–55. [[CrossRef](#)]
 40. Pang, N.; Su, H.; Shi, Y. Project Robust Scheduling Based on the Scattered Buffer Technology. *Appl. Sci.* **2018**, *8*, 541. [[CrossRef](#)]
 41. Deblaere, F.; Demeulemeester, E.; Herroelen, W.; Van de Vonder, S. Robust Resource Allocation Decisions in Resource-Constrained Projects *. *Decis. Sci.* **2007**, *38*, 5–37. [[CrossRef](#)]
 42. Valls, V.; Ballestín, F.; Quintanilla, S. Justification and RCPSP: A Technique That Pays. *Eur. J. Oper. Res.* **2005**, *165*, 375–386. [[CrossRef](#)]