Available online at www.sciencedirect.com
SCIENCE DIRECT®
European Journal of Operational Research 164 (2005) 639–668
EUROPEAN JOURNAL OF OPERATIONAL RESEARCH
www.elsevier.com/locate/dsw

Simulated annealing and tabu search for multi-mode resource-constrained project scheduling with positive discounted cash flows and different payment models

Marek Mika [a,*], Grzegorz Waligóra [a], Jan Węglarz [a,b]

[a] *Institute of Computing Science, Poznań University of Technology, Piotrowo 3A, 60-965 Poznań, Poland*
[b] *Poznań Supercomputing and Networking Center, Noskowskiego 10, 61-704 Poznań, Poland*

Received 1 December 2002; accepted 1 October 2003
Available online 6 May 2004

Abstract

In this paper the multi-mode resource-constrained project scheduling problem with discounted cash flows is considered. A project is represented by an activity-on-node (AoN) network. A positive cash flow is associated with each activity. Four different payment models are considered: lump-sum payment at the completion of the project, payments at activities' completion times, payments at equal time intervals and progress payments. The objective is to maximize the net present value of all cash flows of the project. Local search metaheuristics: simulated annealing and tabu search are proposed to solve this strongly NP-hard problem. A comprehensive computational experiment is described, performed on a set of instances based on standard test problems constructed by the ProGen project generator, where, additionally, the activities' cash flows are generated randomly with the uniform distribution. The metaheuristics are computationally compared, the results are analyzed and discussed and some conclusions are given.

Keywords: Project management and scheduling; Multi-mode resource-constrained project scheduling problem with discounted cash flows; Net present value; Simulated annealing; Tabu search

1. Introduction

In the classical resource-constrained project scheduling problems the project duration (or makespan) is to be minimized. The multi-mode problem (MRCPSP) is a generalized version of the standard problem (RCPSP), where each activity can be executed in one of several modes representing a relation between resource requirements of the activity and its duration. The schedule has to be precedence- and resource-feasible

* Corresponding author.
E-mail addresses: marek.mika@cs.put.poznan.pl (M. Mika), grzegorz.waligora@cs.put.poznan.pl (G. Waligóra), jan.weglarz@cs.put.poznan.pl (J. Węglarz).

0377-2217/$ - see front matter
doi:10.1016/j.ejor.2003.10.053

and no activity may be interrupted. A project can be represented by an activity-on-node (AoN) or an activity-on-arc (AoA) network. In the first representation nodes correspond to activities and arcs to precedence constraints, whereas in the second one, nodes correspond to time events and arcs to activities. The resources can be renewable, non-renewable, doubly constrained, and/or partially renewable, where the renewable resources are limited period-by-period, the non-renewable resources are limited for the entire project, the doubly constrained ones are limited both for each period and for the entire project, and the availability of the partially renewable resources is defined for a specific time interval (a subset of periods). However, under discrete resources, the doubly constrained resources need not be taken into account explicitly since they can be incorporated by properly enlarging the sets of the first two types of resources. The objective is to find an assignment of modes to activities as well as precedence- and resource-feasible starting times of all activities such that the makespan of the project is minimized. The problem is strongly NP-hard being a generalization of the RCPSP. The RCPSP is strongly NP-hard as a generalization of the well-known job shop problem [6]. Moreover, for more than one non-renewable resource the problem of finding a feasible solution of the MRCPSP is already NP-complete [39].

Financial aspects of the problem appear when, generally, a series of cash flows (positive and/or negative) occur over the course of the project. Positive cash flows (i.e. cash inflows) correspond to payments for the execution of activities. Negative cash flows (i.e. cash outflows) include expenditures for labour, equipment, materials, etc. Time value of money is taken into account by discounting the cash flows. The most commonly considered financial objective is the maximization of the net present value (NPV) of all cash flows of the project. The resulting problem is then called the multi-mode resource-constrained project scheduling problem with discounted cash flows (MRCPSPDCF). Let us stress that when the objective of minimizing the makespan is replaced with the objective of maximizing the net present value of cash flows, the optimal schedule is not necessarily the same, i.e. the starting times of some (if not all) activities may be different [5]. Moreover, even if the optimal NPV schedule has the duration equal to an optimal makespan schedule, not all alternative optimal makespan schedules (if there are more than one) will give the maximal NPV. However, if all cash flows are positive, the problem of maximizing the net present value subsumes the makespan minimization version of this problem by summing all cash flows and placing them on the final activity of the project.

In real life situations there are at least two parties involved in the project: the client, who is the owner of the project, and the contractor, whose job is to execute the project. They have to agree the method of payment transferring from the client to the contractor for the execution of the project. The ideal situation for the client would be, of course, a single payment at the end of the project. The contractor, on the other hand, would like to receive the whole payment at the beginning of the project. It should be stressed that in this paper we approach the problem from the contractor's perspective, i.e. we look for solutions maximizing the net present value from the contractor's point of view. There are several types of contracts and payment models considered in the literature. For a comprehensive survey see e.g. [26] or [11].

In this paper we deal with the MRCPSPDCF, where there are only positive cash flows and they are associated with activities, more precisely, a cash inflow (i.e. a payment going from the client to the contractor) is associated with the execution of each activity. This assumption corresponds to a common situation, where the contractor's expenditures do not depend on time but are fixed independently from the project duration, or they are simply covered by the client. As a practical example, we can consider a large software project realized on the client's order (e.g. bank, insurance company, etc.), implemented according to software engineering requirements. A software company (the contractor) has its own hardware (computers), software (operating systems, editors, compilers) and programmers, and bears only fixed expenses connected with its business activity. These expenses do not depend on the realized project. Another example could come from building engineering. A private investor (the client) places an order of house building in a building company (the contractor). The company has its own workers, specialized equipment, transport facilities, etc. The expenses of all materials, energy, fuel and so on are covered by the client. Also in such a case, the contractor's expenditures do not depend on the project. Many similar examples from other areas can be given as well.

We consider four different payment models which are of special interest in practice: lump-sum payment at the completion of the project (LSP), payments at activities' completion times (PAC), payments at equal time intervals (ETI) and progress payments (PP). The LSP model represents the ideal situation for the client—he makes a single payment to the contractor only at the end of the project. However, in general, this shifts the entire financial burden on the contractor, which may not be acceptable in some project environments. In consequence, the two parties often attempt to negotiate a method of payments over the duration of the project, e.g. according to PAC, ETI or PP models. PAC is a very reasonable model, where the contractor gets his payments for successful completion of each activity. In the ETI model the client and the contractor agree about the number of payments over the course of the project. The payments are then made at equal time intervals. A similar situation concerns the PP model, where the payments are also made at regular time intervals, but in this case the two parties agree about the length of this interval, not the number of payments. A typical example is a contract, where at the end of each month the contractor receives his payment for the work accomplished during that month. It should be stressed that we assume the budged of the project to be predetermined, i.e. in each payment model considered the contractor's total payment is the same. The assumed amount of money is transferred by the client to the contractor according to the payment model agreed, but in each case the sum of all payments is identical and equal to the sum of all cash flows associated with the project's activities. As a result, the activities' cash flows do not depend on their execution modes, since the client pays the same regardless of how the contractor realizes the project.

Since, as we have mentioned, the problem is strongly NP-hard, we propose local search metaheuristics: simulated annealing (SA) and tabu search (TS) to approach it. They use the feasible solution representation based on a precedence feasible list of activities and a mode assignment, which is known from the literature for the makespan minimization problem. Our simulated annealing algorithm has already been tested for the latter problem and has produced very good results [32]. The main objectives of this paper are: (i) to examine if the SA algorithm maintains its efficiency when the scheduling criterion is changed, by comparing it on the basis of a computational experiment with another metaheuristic, and (ii) to examine the efficiency of both the metaheuristics for different payment models. Some results concerning a comparison between SA and TS under the PAC model have already been published in [33]. That paper has also shown that the metaheuristics produce significantly better results than a simple random sampling algorithm and, moreover, consume less computational time. Therefore the random approach will be no more taken into consideration in this work.

The paper is organized as follows. In Section 2, we present a review of the literature concerning the resource-constrained project scheduling problems with the NPV criterion. Section 3 defines the considered problem mathematically. In Section 4 the applications of the metaheuristics: SA and TS are described. Section 5 is devoted to the computational experiment and the analysis of the results. Finally, some conclusions and directions for further research are given in Section 6.

2. Literature review

To the best of our knowledge, the idea of maximizing the net present value of the cash flows in a project as a criterion of scheduling activities was introduced by Russell in 1970 [54]. Let us denote such a problem, in general, as a *max-npv* problem. Russell considers a problem without resource constraints, assuming activity-on-arc (AoA) network, where both positive and negative cash flows are associated with events. He presents a non-linear programming problem with continuous discounting in the objective function, and uses a Taylor series approximation to derive a linear objective function. The dual form of this linear programming model can be interpreted as a network flow problem and then the optimal solution procedure is derived. The application of this algorithm is demonstrated using a small example. Grinold in [23] uses the Russell's non-linear model formulation and shows that it can be transformed into a linear programming model directly. By utilizing a special structure of the resulting model, he proposes two optimal solution algorithms for it. The first one solves the problem for a fixed project deadline, whereas the second, parametric one, solves the problem for all possible project deadlines. Again, only a small example is used to demonstrate the presented idea. Elmaghraby and Herroelen in [15] offer an optimal algorithm that builds tree structures in an AoA network in an iterative way and determines proper displacement intervals for trees. A computational study for this algorithm performed for 250 randomly generated projects is reported by Herroelen and Gallens in [25]. But Sepil in [57] shows by means of an elaborate example that this algorithm is flawed and may fail to find an optimal solution. Etgar et al. in [17] consider a problem with an AoA network, where cash flows are associated with events. They assume that the cash flow magnitudes are dependent on the time of realization according to the changing costs of resources over time. They also allow incentive payments for early event occurrences and penalties for late event occurrences. A simulated annealing approach with different implementation strategies is proposed and tested for 168 different problems. The same problem is considered by Shtub and Etgar in [59] who present a branch-and-bound approach to it. Etgar again considers this problem in [16], where he models its special version, assuming that cash flows are linear functions of the event's realization times. The analysis of these linear functions reveals special properties of the problem, which he exploits to develop an effective, optimal solution. De Reyck and Herroelen in [53] consider an activity-on-node (AoN) network with minimal and maximal time lags, where positive and negative cash flows are associated with activities. They present a depth-first branch-and-bound algorithm in which the nodes in the search tree represent the original project network extended with extra precedence relations which resolve a number of resource conflicts using the concept of minimal delaying modes originally developed for the RCPSP. The algorithm is validated on the basis of an extensive computational experiment, performed on a set of randomly generated benchmark problems. Kazaz and Sepil in [34] study a model with progress payments based on time and apply Benders' decomposition to solve this problem. Erengüç et al. in [18] present a problem with an AoA network, where activities can be performed in one out of several modes. Cash flows are associated with both events and activities. The amount of the cash flows connected with activities depends on the mode assignment. Benders' decomposition is used to solve this problem. Schwindt and Zimmermann in [56] consider the unconstrained max-npv problem with general temporal constraints and propose algorithm based on a first-order steepest ascent approach, where the steepest ascent directions are normalized by the supremum norm. The efficiency of this algorithm is demonstrated on the basis of two randomly generated benchmark problem sets consisting of instances with up to 1000 activities. Vanhoucke et al. in [67] study the unconstrained max-npv problem with a fixed project deadline, where each activity has a known deterministic cash flow function that is non-increasing and linear-dependent on the completion time of the corresponding activity. Progress payments and cash outflows occur at the completion times of activities. An extension of an exact recursive algorithm that has been used in solving the max-npv problem with time-independent cash flow functions is proposed. The enumeration procedure, in which the algorithm is embedded, is validated on a randomly generated problem set.

Adding resource constraints to the max-npv problem results in an NP-hard optimization problem. Some recent surveys on resource-constrained project scheduling problems (RCPSP) and RCPSP with discounted cash flows (RCPSPDCF) are given by Icmeli et al. in [31], Elmaghraby in [14], Özdamar and Ulusoy in [48], Herroelen et al. in [26,27], Kolisch and Padman in [40], Brucker et al. in [7], Kimms in [37]. Some authors took the challenge to develop exact algorithms for the RCPSPDCF. Doersch and Patterson in [13] formulate a zero-one integer programming model, where capital is the only scarce resource. The model is solved using standard software and tested with an example consisting of eight activities. Yang et al. in [69] propose a depth-first branch-and-bound algorithm and use it to solve 10 problems with up to 21 activities. They assume an AoN network and cash flows which occur during the performance of each activity. Icmeli and Erengüç in [29] use a branch-and-bound approach to a problem with an AoN network and both positive and negative cash flows associated with the project's activities. This approach is validated on the basis of 50 instances with up to 51 activities from the so-called Patterson's data set [51] as well as 40 problems with up to 32 activities generated using the ProGen test-bed generator [41]. Baroum and Patterson in [4] consider an AoN network with non-negative cash flows associated with the activities. A branch-and-bound procedure is proposed to solve this problem and tested using 15 out of the 110 problems from Patterson's data set. The largest considered problem consists of 51 activities. De Reyck and Herroelen in [53] consider the RCPSPDCF with generalized precedence relations and both positive and negative cash flows associated with the activities. They propose a depth-first branch-and-bound algorithm which is tested by means of a randomly generated benchmark problem set with up to 50 activities.

There are plenty of papers concerning heuristic approaches to the RCPSPDCF. The first heuristic applied to the RCPSP with discounted cash flows was developed by Russell in [55]. He assumes AoA networks, where positive and negative cash flows are associated with events. Six heuristics: the random rule (selection of the best out of 50 randomly generated solutions), the minimum slack rule, the minimum latest finishing time rule and three heuristics based on the optimal results for the unconstrained max-npv problem are proposed and tested on 70 different problems with up to 1461 activities. Smith-Daniels and Aquilano in [61] assume AoN networks, where cash outflows occur at the beginning of each activity and a single lump-sum payment is received at the completion of the project. They propose a heuristic building late-start schedules and examine it by means of 550 test problems with up to 50 activities. Padman and Smith-Daniels in [49] present eight greedy heuristics embedded in a single-pass, forward algorithm that uses information from a relaxed (resource-unconstrained) optimization model. A computational experiment is performed with 1440 problems with up to 110 activities. This work has been developed further by Smith-Daniels et al. in [62] and Padman et al. in [50] who present a few additional heuristics as well as study problems with up to 230 and up to 1000 activities, correspondingly. Icmeli and Erengüç in [28] use two tabu search adaptations to solve the RCPSPDCF with AoN networks, where cash flows are associated with activities and a penalty occur if the project is completed later than a given due date. Their approaches were tested on 50 problems derived from Patterson's data set and compared to upper bounds obtained from a linear programming relaxation of the RCPSPDC and to solutions obtained by the minimum slack heuristic. Yang et al. in [70] propose nine stochastic scheduling rules and show on the basis of a comprehensive computational experiment consisting of 1440 problems with up to 20 activities that simulated annealing, the Rank Positional Weight and Discounted Cumulative Cash Flow Weight rules perform very well, generating high net present values. Baroum and Patterson in [3] describe several priority rule heuristics and compare them on the basis of computational experiment, where the largest data set consists of 50 activities. The best performing heuristic is a multi-pass one. Pinder and Marucheck in [52] develop 10 new scheduling heuristics and compare them with seven well-known rules. They assume AoN networks with cash flows associated with activities and show on the basis of a computational experiment, performed using a data set with up to 200 activities, that two of the new heuristics are significantly better than the standard heuristics. Sepil and Ortaç in [58] consider a problem with progress payments for which they propose three different heuristic rules. The performance of these heuristics is analyzed through a computational experiment with 108 scheduling conditions and with up to 200 activities. Zhu and Padman in [71] propose a tabu search approach. Dayanand and Padman in [10] examine the problem of simultaneously determining the amount, location and timing of progress payments. They present several models and their solutions for an example problem. In [12] they propose a two-stage heuristic for the same problem, where simulated annealing is used in the first stage to determine a set of payments. In the second stage, activities are rescheduled to improve the NPV of the project. The authors compare the performance of this general purpose heuristic with other problem-dependent heuristics from the literature. The results indicate that the simulated annealing metaheuristic significantly outperforms the parameter-based heuristics. The RCPSPDCF with general temporal constraints given by minimum and maximum time lags between activities is attacked by Neumann and Zimmermann in [45] who develop various algorithms to solve this problem. Vanhoucke et al. in [68] consider the RCPSPDCF with a fixed deadline and both positive and negative cash flows. Progress payment and cash outflows occur at the completion of activities. For this problem a depth-first branch-and-bound method is proposed and validated on two sets of benchmark instances. Kimms in [36] derives tight upper bounds for the RCPSPDCF on the basis of the Lagrangian relaxation of the resource constraints. He also uses this approach as a basis for a heuristic and shows that the heuristic as well as the cash flow weight heuristic proposed by Baroum and Patterson in [3] yield solutions very close to optimum. Furthermore, he discusses the proper choice of a test-bed and emphasizes that discount rates must be carefully chosen to give realistic instances.

The multi-mode version of the RCPSPDCF has been considered in several articles. Sung and Lim in [64] study a problem with positive and negative cash flows and availability constraints on capital and renewable resources. They propose a two-phase heuristic and validate it by means of a computational experiment. Icmeli and Erengüç in [30] present a problem, where the activities' durations can be reduced from their normal durations by allocating more resources. They propose a heuristic procedure with embedded priority rules and compare obtained results with tight upper bounds obtained using the Lagrangian relaxation. Özdamar and Dündar in [47] consider housing projects, where an initial capital covers activity expenditures in the starting phase of the project and then customers, who arrive randomly over the project span, provide the necessary funds for continuation. Capital is considered as a limited non-renewable resource which is reduced by activity expenditures and augmented by the sales of flats. The total cost of an activity is fixed irrespective of its operating mode. They propose a flexible heuristic algorithm for solving the capital-constrained mode selection problem, where there exist general precedence relationships among activities and the magnitude of precedence lags depend on the specific activity mode selected. The algorithm is tested using a typical housing project with real data and also by using hypothetical test problems. Özdamar in [46] considers situations in the housing industry, where the contractor is the owner of the project. In this case, the contractor starts with an initial capital to cover the activities' expenditures and then capital is augmented by the sale of flats, which take place randomly over the progress of the project. In this risky environment, the contractor has to decide on the rate of expenditure at each decision time in order to maintain a positive cash balance. Hence, activities are executed in multiple performing modes with different durations and the same total cost. A heuristic to construct and re-construct schedules during the progress of the project is proposed with the aim of maximizing NPV while completing the project on time. The heuristic incorporates dynamic mode selection criteria which change adaptively according to the current status of the project. Computational experiments demonstrate that the heuristic provides satisfactory results regarding the feasibility of the schedules with respect to the project due date and the non-renewable resource constraints. Ulusoy and Cebelli in [65] consider a problem, where the goals of the contractor and the client are joined in one model. They used a double-loop genetic algorithm to find an equitable solution, which is defined as such in which both the contractor and the client deviate from their respective ideal solutions by an equal percentage. The ideal solutions for the contractor and the client result from having a lump-sum payment at the start and at the end of the project, respectively. A set of 93 problems from the literature are solved and some computational results are reported. Ulusoy et al. in [66] present the MRCPSPDCF with renewable, non-renewable and doubly constrained resources. Positive and negative cash flows are associated with events and/or activities, depending on the considered model. Four payment models are considered: lump-sum payment at the terminal event, payment at fixed event nodes, payments at prespecified time points and progress payments. A genetic algorithm with a special crossover operator is used to solve the considered problem. To the best of our knowledge, no exact procedure for the MRCPSPDCF exists.

3. Problem formulation

We consider the multi-mode resource-constrained project scheduling problem with positive discounted cash flows (MRCPSPDCF) which can be formulated as follows. A project consisting of n activities is represented by an activity-on-node (AoN) network $G = (J, E)$, $|J| = n$, where nodes and arcs correspond to activities and precedence constraints between activities, respectively. No activity may be started before all its predecessors are finished. Nodes in graph G are numerically numbered, i.e. an activity has always a higher number than all its predecessors.

Each activity j, $j = 1, \dots, n$, has to be executed in one of M_j modes. The activities are non-preemtable and a mode chosen for an activity may not be changed (i.e. an activity j, $j = 1, \dots, n$, started in mode m, $m \in 1, \dots, M_j$, must be completed in mode m without preemption). The duration of activity j executed in mode m is d_{jm}. A positive cash flow CF_j is associated with the execution of each activity j.

We assume that there are R renewable and N non-renewable resources. The number of available units of renewable resource k, $k = 1, \dots, R$, is R_k^ρ and the number of available units of non-renewable resource l, $l = 1, \dots, N$, is R_l^ν. Each activity j executed in mode m requires for its processing r_{jkm}^ρ units of renewable resource $k, k = 1, \dots, R$, and consumes r_{jlm}^ν units of non-renewable resource l, $l = 1, \dots, N$. We assume that all activities and resources are available at the start of the project.

When dealing with the NPV criterion time value of money is taken into account by discounting the cash flows. The value of an amount of money is a function of the time of receipt or disbursement of cash. A dollar received today is more valuable than a dollar to be received in some time in the future, since the today's dollar can be invested immediately. In order to calculate the value of NPV, a *discount rate* α has to be chosen, which represents the return following from investing in the project rather than e.g. in securities. Then the *discount factor* $(1 + \alpha)^{-T}$ denotes the present value of a dollar to be received at the end of period T using a discount rate α.

The objective of the MRCPSPDCF is to find an assignment of modes to activities as well as precedence- and resource-feasible starting times for all activities such that the net present value of the project is maximized.

All the MRCPSPDCF parameters are summarized below and, excluding graph G, set P_j and the discount rate α, they are assumed to be integers:

n	number of activities
G	acyclic digraph representing the project
M_j	number of modes of activity j, $j = 1, \dots, n$
d_{jm}	duration of activity j executed in mode m, $m = 1, \dots, M_j$
CF_j	cash flow associated with activity j
ST_j	starting time of activity j
FT_j	finishing time of activity j
EF_j	earliest finishing time of activity j
LF_j	latest finishing time of activity j
P_j	set of all predecessors of activity j
R	number of renewable resources
N	number of non-renewable resources
R_k^ρ	number of available units of renewable resource k, $k = 1, \dots, R$
R_l^ν	number of available units of non-renewable resource l, $l = 1, \dots, N$
r_{jkm}^ρ	number of units of renewable resource k required by activity j executed in mode m
r_{jlm}^ν	number of units of non-renewable resource l consumed by activity j executed in mode m
α	discount rate
TH	time horizon of the project (the upper bound of the project makespan given by the sum of maximal durations of activities).

Using the above notation, the MRCPSPDCF can be formulated as the following mathematical programming problem:

$$\text{maximize} \quad \sum_{j=1}^{n}\sum_{t=\mathrm{EF}_j}^{\mathrm{LF}_j}\frac{\mathrm{CF}_j}{(1+\alpha)^t}x_{jmt}$$

$$\text{subject to:} \quad 1. \quad \sum_{m=1}^{|M_j|}\sum_{t=\mathrm{EF}_j}^{\mathrm{LF}_j}x_{jmt}=1, \quad j=1,\ldots,n,$$

$$2. \quad \sum_{m=1}^{|M_h|}\sum_{t=\mathrm{EF}_h}^{\mathrm{LF}_h}t\cdot x_{hmt}\leqslant\sum_{m=1}^{|M_j|}\sum_{t=\mathrm{EF}_j}^{\mathrm{LF}_j}(t-d_{jm})\cdot x_{jmt}, \quad j=1,\ldots,n; \; h\in P_j,$$

$$3. \quad \sum_{j=1}^{n}\sum_{m=1}^{|M_j|}r^{\rho}_{jkm}\sum_{b=t}^{t+d_{jm}-1}x_{jmb}\leqslant R^{\rho}_k \quad k=1,\ldots,R; \; t=1,\ldots,\mathrm{TH},$$

$$4. \quad \sum_{j=1}^{n}\sum_{m=1}^{|M_j|}r^{v}_{jlm}\sum_{t=\mathrm{EF}_j}^{\mathrm{LF}_j}x_{jmt}\leqslant R^{v}_l \quad l=1,\ldots,N,$$

$$5. \quad x_{jmt}\in\{0,1\}, \quad j=1,\ldots,n; \; m\in M_j; \; t=1,\ldots,\mathrm{TH},$$

where x_{jmt} is a decision variable which is equal to 1 if and only if activity j is performed in mode m and finished at time t.

Constraint set 1 secures that each activity is assigned exactly one mode and exactly one finishing time. Precedence feasibility is maintained by constraint set 2. Constraint sets 3 and 4 take care of the renewable and non-renewable resource limitations, respectively. And finally, constraint set 5 define the binary status of the decision variables.

The way of calculating the value of NPV depends on the payment model considered and will be presented below.

3.1. Lump-sum payment

Lump-sum payment (LSP) is one of the payment models most commonly considered in the literature. It represents the ideal situation for the client since in this case the whole payment is paid by the client to the contractor at the moment of the successful completion of the project. The total payment is calculated as the sum of the cash flows of all activities. As a result, the value of NPV is given by the following formula:

$$\mathrm{NPV}=\sum_{j=1}^{n}\mathrm{CF}_j(1+\alpha)^{-C_{\max}}, \tag{1}$$

where $C_{\max}$ is the completion time of the project (i.e. the project makespan).

It is easy to see that under the LSP model the maximization of NPV is equivalent to the minimization of $C_{\max}$. Thus, in fact, we obtain the MRCPSP problem here.

3.2. Payments at activities' completion times

Payments at activities' completion times (PAC) is a very practical payment model, where the client pays the contractor for the completion of each activity of the project. Once an activity is finished, the contractor gets the amount of money equal to the cash flow associated with this activity. In this case the value of NPV is given by the following formula:

$$\mathrm{NPV} = \sum_{j=1}^{n} \mathrm{CF}_j (1+\alpha)^{-\mathrm{FT}_j}. \tag{2}$$

3.3. Equal time intervals

In the equal time intervals (ETI) model the client makes H payments for the project. The first $H-1$ payments are scheduled at equal time intervals over the course of the project and the last payment is made at the completion of the project. In this case the value of NPV can be given by

$$\mathrm{NPV} = \sum_{p=1}^{H} P_p (1+\alpha)^{-T_p}, \tag{3}$$

where P_p is the payment at payment point p, $p = 1, 2, \ldots, H$, and T_p is the occurrence time of payment point p. Of course, for the last payment point $H, T_H = C_{\max}$. In order to compute the payments P_p at successive payment points, partial cash flows (linearly proportional to activities' durations) of all the activities executed in the considered interval (i.e. since the last payment point) are summed up.

E.g. let five activities be executed in the considered interval as it is shown in Fig. 1. The length of the interval equals 8 time units and within this interval 1 unit of activity #1 is executed, 5 units of activity #2, 2 units of activity #3, 3 units of activity #4 and 5 units of activity #5. If activities' durations are: 4, 5, 4, 8 and 10, whereas their cash flows: 254, 197, 602, 490 and 936, respectively, then the payment at point p is calculated as:

$$P_p = (1/4) \cdot 254 + (5/5) \cdot 197 + (2/4) \cdot 602 + (3/8) \cdot 490 + (5/10) \cdot 936 = 1213.25.$$

3.4. Progress payments

In the progress payment (PP) model the client makes the payments to the contractor at regular time intervals until the project is completed (e.g. at the end of each month the contractor may receive a payment for the work accomplished during that month). The significant difference in comparison to the ETI model is that in this case the length of the time interval between payments does not depend on the project duration and, in consequence, the number of payments is not known a priori. The value of NPV is given by a formula similar to formula (3) but here the values of T_p are multiples of the assumed interval length T (apart from the last T_p equal to $C_{\max}$):

$$\mathrm{NPV} = \sum_{p=1}^{H-1} P_p (1+\alpha)^{-p \cdot T} + P_H (1+\alpha)^{-C_{\max}}, \tag{4}$$

where H is the smallest integer greater or equal to $C_{\max}/T$.

The payments P_p are calculated in the same way as in the ETI model.

Following the classification given by Herroelen et al. in [24], the problems considered in this paper can be classified as:

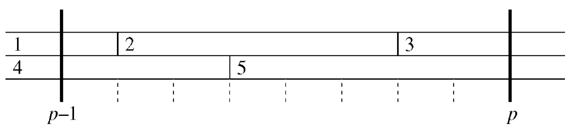

Fig. 1. Fragment of a schedule.

$m, 1T/cpm, mu, c_j^+/npv$ for the LSP and PAC models, and

$m, 1T/cpm, mu, per/npv$ for the ETI and PP models.

4. Metaheuristics

4.1. Common features

In this section we describe these elements of the metaheuristic algorithms which are common for both the approaches used.

4.1.1. Solution representation

A feasible solution is represented by two n-element lists. The first one is a precedence-feasible permutation of activities, in which each activity j, $j = 1, \dots, n$, has to occur after all its predecessors and before all its successors. This structure is called the *activity list*. The second one is a list of execution modes for all activities and is called the *mode assignment*. The jth element of this list defines the execution mode of activity j. This representation has been used in many previous algorithms for the MRCPSP.

Having got a feasible solution represented by the two lists described above, the starting times of all activities are then defined by using the serial SGS (Schedule Generation Scheme) decoding rule [35]. Since under positive cash flows the NPV criterion is a regular performance measure [39], i.e. a measure which is non-decreasing in activity completion times [9], we may use the serial SGS rule to construct the schedule. It is easy to see that in the considered case any schedule in which an activity is delayed, while it is possible for it to be scheduled earlier, is an inferior schedule, because a higher value of NPV would result from advancing the occurrence of the delayed activity. As a result, there is no danger of omitting an optimal schedule by using the serial SGS here.

4.1.2. Preprocessing

In order to reduce the search space and adapt the project data to the implementations of the metaheuristics, the preprocessing procedure introduced by Sprecher et al. in [63] is used. After the execution of this iterative procedure all non-executable modes, inefficient modes as well as redundant non-renewable resources are deleted.

4.1.3. Objective function

The objective is to maximize the net present value. In order to avoid problems with solutions not feasible with respect to the non-renewable resources (in many instances it is not easy to find any feasible solution because of the non-renewable resource constraints), we use a penalty function, i.e. we treat them as feasible solutions, provided that the value of the objective function for such a solution is much worse than for any feasible solution. Moreover, this value depends on the degree of infeasibility, i.e. the more a solution exceeds the non-renewable resource constraints (counted in units), the worse value of the objective function it gets. To this end, for each solution s we calculate a *Solution Feasibility Test* function $\mathrm{SFT}(s)$ according to the following formula:

$$\mathrm{SFT}(s) = \sum_{l=1}^{N} \max\left\{0, \sum_{j=1}^{n} r_{jml}^{\nu} - R_l^{\nu}\right\}. \tag{5}$$

Solution s is feasible with respect to non-renewable resources if and only if $\mathrm{SFT}(s) = 0$.

Finally, depending on the payment model, we define the penalty function for each solution s as

$$\mathrm{NPV}(s) = \begin{cases} (1) \text{ or } (2) \text{ or } (3) \text{ or } (4) & \text{if } \mathrm{SFT}(s) = 0, \\ \sum_{j=1}^{n} \mathrm{CF}_j (1+\alpha)^{-[\mathrm{TH}+\mathrm{SFT}(s)]} & \text{otherwise}, \end{cases} \tag{6}$$

where TH is the time horizon of the project (the upper bound on the project makespan).

4.1.4. Starting solution

A starting solution for each instance is generated by setting all activities on the activity list in an ascending order that follows from the ordering of nodes in the precedence relation graph, and executing all activities in their first modes. This procedure has been commonly used in local search algorithms for the MRCPSP before.

4.1.5. Stop criterion

The stop criterion has been defined as an assumed number of visited solutions, i.e. an assumed number of the objective function calculations.

4.2. Simulated annealing

Simulated annealing is a well-known local search metaheuristic, which belongs to a class of the threshold algorithms as presented by Aarts et al. in [2], and can be viewed as a special case of the First Fit Strategy, where the next solution is accepted with a certain probability. SA is based on the Monte Carlo method introduced by Metropolis et al. in [44]. This idea was originally used to simulate a physical annealing process and was applied to combinatorial optimization for the first time in the 1980s independently by Kirkpatrick et al. [38] and Černy [8]. For the presented adaptation of the SA algorithm a homogeneous version is used [43]. In order to develop the presented implementation of SA for the considered problem a number of decisions have to be made.

4.2.1. Neighbourhood

Neighbour solutions are generated from the current one using one of the following three operators:

- *Activity shift*–operates on the activity list (AL) in the following way:
  - one activity j is randomly chosen on AL;
  - the nearest predecessor p and the nearest successor s of activity j are found on AL;
  - a position x between activities p and s is randomly chosen;
  - activity j is moved to position x.
- *Mode change*–operates on the mode assignment (MA) in the following way:
  - a position y on MA is randomly chosen;
  - the mode in position y is changed to another one randomly chosen, if possible.
- *Combined move*–operates simultaneously on both AL and MA and is a composition of the activity shift and the mode change.

For each transition the above-mentioned operators are chosen randomly with a certain probability which depends on the instance of the considered problem. The probability of a combined move is set at a fixed value of 0.1. The probability of an activity shift depends on the density of graph G and is given by the following formula:

$$P_{AS} = \left(\frac{1}{2} - \frac{|E| + |A|}{n \cdot (n-1)} \right), \tag{7}$$

where $|E|$ is the cardinality of the set of precedence constraints (i.e. the set of arcs of graph G), $|A|$ is the cardinality of set A of arcs of the transitive closure of graph G, and n is the number of the project's activities. The probability of the mode change $P_{MCh} = 0.9 - P_{AS}$. The neighbourhood generation mechanism is used during the cooling process to generate the next solution from the current one.

4.2.2. Cooling scheme

The adaptive cooling scheme known as "polynomial-time", described by Aarts and Korst in [1], is used to control the cooling process of the SA algorithm. The only exception introduced into this implementation of SA is the stop criterion, which is set at a fixed number of visited solutions.

The initial value T_0 of the control parameter is calculated during the initialization phase from the following equation:

$$T_0 = \frac{\overline{\Delta f}^{(+)}}{\ln\left(\frac{m_2}{m_2 \cdot \chi_0 - m_1(1-\chi_0)}\right)}, \tag{8}$$

where χ_0 is the initial acceptance ratio (the assumed proportion between transitions accepted and all the transitions generated for T_0); m_1 is the number of cost-non-decreasing transitions and m_2 is the number of cost-decreasing transitions from among m_0 trial transitions generated to determine the initial value of the control parameter. Of course, $m_0 = m_1 + m_2$. $\overline{\Delta f}^{(+)}$ is the average difference in cost over the m_2 cost-decreasing transitions. It is assumed that $\chi_0 = 0.95$ and $m_0 = 50$.

The next value T_{k+1} of the control parameter calculated during the cooling process depends on the mean and the standard deviation σ_{T_k} for the values of the objective function for the kth Markov chain. The value T_{k+1} is calculated using the following formula:

$$T_{k+1} = \frac{T_k}{1 + \frac{T_k \cdot \ln(1+\delta)}{3 \cdot \sigma_{T_k}}}, \quad k = 0, 1, \ldots, \tag{9}$$

where δ is a real number denominating the distance parameter. Usually, smaller values of this parameter lead to better solutions but also increase the computation time. It is assumed in our implementation that $\delta = 0.5$.

The length of the Markov chain in the homogeneous version of SA determines the number of transitions for a given value of the control parameter. It is assumed that this value is constant and depends on the size of the problem. The length of kth Markov chain is calculated according to the following rule:

$$L_k = L = 5 \cdot n, \quad k = 0, 1, \ldots, \tag{10}$$

where n is the number of activities.

4.3. Tabu search

Tabu search is a metastrategy based on neighbourhood search with overcoming local optimality. It works in a deterministic way, trying to model human memory processes. Memory is implemented by the implicit recording of previously seen solutions, using simple but effective data structures. Tabu search was originally developed by Glover in [19–21], and a comprehensive report of the basic concepts and recent developments is given by Glover and Laguna in [22].

4.3.1. Neighbourhood

Similarly as it was for simulated annealing, neighbours of the current solution are generated by swapping activities on the activity list and changing modes on the mode assignment. More precisely, in the case of TS there are two operators used in the neighbourhood generation mechanism:

- *Activity swap*–swaps each two activities on the activity list that may be swapped without violating the precedence constraints; as a result, a set of neighbouring solutions is created each of them differing from the current solution in two positions of the activity list.
- *Mode change*–changes the execution mode of each activity to every other executable mode for this activity; as a result a set of neighbouring solutions is created each of them differing from the current solution in one position of the mode assignment.

It is easy to see that using the above operators may make the neighbourhood be quite large, especially when there are not many precedence constraints. However, this neighbourhood generation mechanism has proved to control the search process very effectively at the stage of some preliminary experiments.

4.3.2. Moves

Considering the neighbourhood generation mechanism described above, moves are represented by the following sets of attributes:

- For the activity swap operator it is a triple

$$(\textit{position number on the activity list},\ \textit{activity replaced},\ \textit{activity inserted}),$$

where these values concern the activity with the smaller position number on the activity list, i.e. occurring closer to beginning of the list.

E.g. if activity 2 in position #3 is swapped with activity 6 in position #8 then the attributes of the performed move are the following:

$$(3, 2, 6),$$

which means that in position #3 activity 2 has been replaced by activity 6.

In consequence, the reverse move should forbid activity 2 to go back into position #3 and has the form of the following couple:

$$(3, 2).$$

This move will be added to the tabu list.

- For the mode change operator it is a triple

$$(\textit{position number on the activity list},\ \textit{mode replaced},\ \textit{mode inserted})$$

E.g. if mode 2 in position #14 is changed to mode 1, then the attributes of the performed move are the following:

$$(14, 2, 1).$$

In consequence, the reverse move should forbid mode 2 to appear again in position #14 and has the form of the following couple:

$$(14, 2).$$

This move will be added to the tabu list.

4.3.3. Tabu list

The tabu list is managed according to the *Tabu Navigation Method* [60]. There is only one list, so-called *Tabu List* (TL), which is a queue of a given length. Whenever a move is performed, its reverse is added to the TL in order to avoid going back to a solution already visited. The oldest existing move is removed from the front of the list (according to the FIFO policy). All moves existing on the TL are tabu. However, the tabu status of a move may be cancelled according to so-called *aspiration criterion*. The important parameter in the TNM is the length of the TL. In our implementation it is set at 7.

In the case of the TNM, it may happen that a move existing on the TL does not allow to reach a solution which has not been visited yet. In order to avoid a situation where a good solution is overlooked, an aspiration criterion was applied which allows the algorithm to move to a tabu solution (perform a tabu move) if this solution is better than the best found so far. If this is the case, the tabu move performed is removed from the TL (the oldest move stays on the TL) and the reverse move is added to the end of the list. Of course, performing such a move cannot lead to a solution visited in the past because in this case the obtained objective function value would already have been known.

4.3.4. Restarting

In case there are no admissible solutions in a neighbourhood (it may possibly happen because of the tabu list restrictions), the algorithm has to restart the search process in order to visit an assumed number of solutions. Additionally, the restarting mechanism is also used when an assumed number of iterations have been performed without improving the objective function. This number has been set at 20. If 20 iterations with no improvement are performed the algorithm restarts since further search in that solution region is considered hopeless.

In order to restart the search process from a different solution, it is generated randomly, i.e. the activity list is generated in the same way as described in point 4.1.4 but the mode assignment is generated randomly. This assures that the algorithm will start from another solution which is, of course, important taking into account its deterministic character.

Thus, after the restart the following actions are performed:

- a new starting solution is generated as it has been described earlier;
- the tabu list is cleared;
- the counter of iterations without improvement is reset.

5. Computational experiment

In this section we present the results of a computational experiment concerning the implementations of the two metaheuristics for the considered MRCPSPDCF. The implementations were coded and compiled in C++ and ran on an SGI ORIGIN 3800 machine with 64 RISC R12000 processors and 57.6 Gflops overall computing power, installed in the Poznań Supercomputing and Networking Center.

We used a set of benchmark problem instances generated using the project generator ProGen developed by Kolisch et al. in [42]. The files with the mentioned instances as well as the code of ProGen itself are available in the project scheduling problem library PSPLIB from the ftp service of the University of Kiel (*ftp://ftp.bwl.uni-kiel.de/pub/operations-research/psplib*). For more details concerning ProGen see [41].

We used benchmark sets for problems with 10, 12, 14, 16, 18, 20 and 30 non-dummy activities. All non-dummy activities may be executed in one of three modes. There are two renewable and two non-renewable resources. The duration of activity j in mode m varies from 1 to 10. For each problem size a set of 640 instances was generated, but for some instances no feasible solution exists and therefore these instances will be excluded from further consideration.

In Table 1 we present the number of instances for which at least one feasible solution exists.

For each instance from the PSPLIB library we have generated cash flows of all activities from the interval (0; 1000] with the uniform distribution.

Table 1
PSPLIB–the number of instances

n	10	12	14	16	18	20	30
Number of instances	536	547	551	550	552	554	552

Table 2
Stop criterion

n	10	12	14	16	18	20	30
Number of solutions	120 000	144 000	168 000	192 000	216 000	240 000	360 000

In order to ensure a comparable computational effort devoted to both the metaheuristics, the stop criterion has been defined as a number of solutions visited. In Table 2 we present the assumed number of visited solutions defining the stop criterion.

Below we present the results of the experiment for all the four payment models considered. In each case the following numbers are shown:

- #–the number of instances for which the algorithm found a solution equal to the best solution known (i.e. best solution found by either of the algorithms);
- #b–the number of instances for which the algorithm found a solution better than the other algorithm;
- AAD–the average absolute deviation from the best solution known;
- MAD–the maximal absolute deviation from the best solution known;
- ARD–the average relative deviation from the best solution known;
- MRD–the maximal relative deviation from the best solution known;
- CPU–the average computational time of the algorithm.

In order to complete the specification of the instances, we need to define α–the discount rate per period. In our experiments we assume that one period corresponds to one month.

For the LSP and PAC models the experiment was performed for five different values of the discount rate. We assumed that $\alpha \in \{0.01, 0.02, 0.03, 0.05, 0.1\}$ and that it is constant over the entire planning horizon. These values were also used for the ETI and PP models. The results are presented in Tables 3 and 4.

For the ETI model we additionally assumed different values of parameter H. The experiment was performed for $H \in \{2, 3, 4, 5\}$. The results are presented in Table 5.

For the PP model we assumed different values of the interval length T. The selection of these values was based on the optimal values of the makespan for the instances used. For each number of activities, the makespan of optimal solutions for the PSPLIB library instances varies from 8 to 62. We assumed the payment interval length $T \in \{3, 4, 6, 12\}$ in our experiment, which corresponds to payments made every 3, 4, 6 and 12 months. Of course, the cases where this length equals or exceeds the project duration ($T \geqslant C_{\max}$), lead directly the LSP model. The results are presented in Table 6.

Taking into account all the problem parameters assumed in the experiment, as well as the number of instances from the PSPLIB library for a given number of activities, it can be noticed that each of the algorithms tested has solved exactly 192 100 instances of the considered problems. Analyzing all the results obtained, the following observations can be made.

As it has been mentioned before, for the LSP model both the metaheuristics try to minimize the project makespan. In the case of TS, for all the values of the discount rate α always the same best solution has been found. It follows from the deterministic nature of this algorithm—since under the minimization of the makespan the value of α has no influence on the quality relation between solutions, tabu search will always

Table 3
Results of the experiment for the LSP model

n	α	TS							SA						
		#	#b	AAD	MAD	ARD [%]	MRD [%]	CPU [seconds]	#	#b	AAD	MAD	ARD [%]	MRD [%]	CPU [seconds]
10	0.01	529	72	0.69	88.04	0.02	1.97	0.79	464	7	11.24	274.71	0.28	8.57	0.96
	0.02	529	76	0.97	134.66	0.03	3.88	0.79	460	7	18.29	474.08	0.56	16.32	0.96
	0.03	530	85	0.85	109.25	0.03	5.74	0.79	451	6	22.75	614.61	0.85	18.69	0.96
	0.05	529	95	1.28	185.87	0.07	9.30	0.79	441	7	28.40	768.81	1.58	38.61	0.96
	0.1	531	100	0.42	94.23	0.08	17.36	0.79	436	5	26.19	942.53	3.26	53.35	0.96
12	0.01	512	57	3.18	112.58	0.07	1.97	1.13	490	35	9.44	316.87	0.20	6.73	1.36
	0.02	516	61	4.55	151.89	0.13	3.88	1.13	486	31	14.61	481.24	0.43	12.94	1.37
	0.03	517	65	4.93	187.59	0.17	5.74	1.13	482	30	19.17	1000.84	0.70	23.36	1.36
	0.05	525	76	4.42	232.11	0.22	9.30	1.13	471	22	23.05	580.36	1.31	32.32	1.37
	0.1	526	91	2.59	146.27	0.36	17.36	1.13	456	21	19.88	619.42	2.87	43.55	1.36
14	0.01	464	38	9.43	177.40	0.18	2.94	1.58	513	87	8.60	388.03	0.17	7.65	1.86
	0.02	475	45	13.13	264.04	0.30	5.77	1.58	506	76	16.41	619.26	0.44	16.32	1.85
	0.03	474	55	15.67	295.60	0.46	8.49	1.58	496	77	20.09	687.56	0.73	23.36	1.85
	0.05	480	66	15.70	303.02	0.71	17.73	1.58	485	71	19.94	664.44	1.21	32.32	1.85
	0.1	501	101	7.62	264.58	0.92	31.70	1.58	450	50	17.75	515.18	2.94	43.55	1.86
16	0.01	401	29	22.29	254.53	0.36	3.90	2.11	521	149	4.97	375.17	0.09	6.73	2.49
	0.02	410	37	31.37	366.02	0.66	7.62	2.11	513	140	8.82	511.25	0.21	12.94	2.49
	0.03	415	39	34.41	396.08	0.93	11.15	2.11	511	135	12.21	810.88	0.41	35.81	2.48
	0.05	416	42	35.34	386.47	1.58	17.73	2.11	508	134	12.44	416.82	0.68	28.93	2.48
	0.1	429	69	19.24	282.52	2.69	31.70	2.11	481	121	13.46	695.42	2.16	48.68	2.48
18	0.01	357	19	36.63	354.58	0.54	4.85	2.77	533	195	6.66	713.17	0.10	11.26	3.20
	0.02	371	30	50.54	562.69	0.97	7.62	2.77	522	181	12.14	877.37	0.25	21.15	3.19
	0.03	374	33	55.32	671.31	1.37	11.15	2.77	519	178	13.86	814.30	0.40	29.86	3.19
	0.05	378	32	51.75	712.70	2.19	17.73	2.77	520	174	10.47	506.37	0.57	17.73	3.19
	0.1	391	43	26.48	399.02	3.74	31.70	2.77	509	161	8.27	623.85	1.71	73.67	3.18
20	0.01	301	17	60.42	418.31	0.82	5.80	3.51	537	253	3.82	326.83	0.05	4.85	4.06
	0.02	309	26	83.36	616.61	1.53	9.43	3.51	528	245	9.10	572.20	0.18	11.20	4.06
	0.03	317	29	86.62	516.92	2.10	13.74	3.51	525	237	11.15	933.42	0.33	39.50	4.06
	0.05	310	24	82.53	564.60	3.52	25.38	3.51	530	244	8.17	669.79	0.41	35.54	4.03
	0.1	323	36	40.51	406.69	6.11	43.55	3.51	518	231	6.96	341.02	1.11	37.91	4.03
30	0.01	224	1	216.88	935.58	2.11	10.37	9.19	551	328	2.05	1125.63	0.02	12.13	10.24
	0.02	222	0	283.67	1246.04	3.94	17.97	9.19	552	330	0.00	0.00	0.00	0.00	10.21
	0.03	223	1	295.58	1669.57	5.83	29.86	9.19	551	329	0.17	96.10	0.01	2.91	10.19
	0.05	225	1	235.65	1193.08	9.05	44.32	9.19	551	327	0.34	188.34	0.02	13.62	10.18
	0.1	231	2	92.10	664.23	15.65	68.14	9.19	550	321	0.09	43.29	0.08	24.87	10.20

Table 4
Results of the experiment for the PAC model

n	α	TS							SA						
		#	#b	AAD	MAD	ARD [%]	MRD [%]	CPU [seconds]	#	#b	AAD	MAD	ARD [%]	MRD [%]	CPU [seconds]
10	0.01	406	172	3.29	122.25	0.07	2.93	1.09	364	130	14.67	321.15	0.31	5.55	1.28
	0.02	408	190	6.16	215.96	0.15	5.71	1.09	346	128	29.83	561.89	0.71	11.57	1.28
	0.03	412	205	7.84	286.79	0.20	8.30	1.09	331	124	46.61	699.27	1.19	16.80	1.28
	0.05	418	230	11.96	384.12	0.36	13.09	1.09	306	118	72.13	915.78	2.14	22.63	1.28
	0.1	440	310	13.48	471.20	0.54	22.14	1.09	226	96	106.37	1026.39	4.34	38.60	1.28
12	0.01	371	159	4.34	67.22	0.08	1.89	1.58	388	176	18.66	372.50	0.35	6.42	1.84
	0.02	374	195	7.64	121.01	0.16	3.85	1.58	352	173	41.42	844.40	0.85	13.18	1.84
	0.03	382	207	9.53	163.74	0.21	5.86	1.58	340	165	60.70	784.11	1.37	20.76	1.84
	0.05	401	253	11.60	223.51	0.31	9.94	1.58	294	146	93.80	1124.31	2.52	25.45	1.83
	0.1	416	303	14.81	281.11	0.56	19.51	1.58	244	131	146.77	1688.59	5.36	47.46	1.83
14	0.01	279	180	8.46	131.93	0.14	2.26	2.19	371	272	20.85	485.26	0.34	7.09	2.51
	0.02	296	196	14.13	228.73	0.26	4.43	2.19	355	255	42.47	754.76	0.77	12.69	2.50
	0.03	332	233	17.28	298.85	0.35	7.71	2.19	318	219	67.50	1029.87	1.36	18.10	2.50
	0.05	356	269	19.82	375.96	0.47	11.36	2.19	282	195	115.51	1459.89	2.81	30.87	2.49
	0.1	385	320	21.98	376.95	0.71	13.33	2.19	231	166	174.90	2140.15	5.87	44.38	2.48
16	0.01	223	179	13.97	112.32	0.20	1.62	2.91	371	327	23.63	504.35	0.33	5.65	3.34
	0.02	234	193	23.83	192.09	0.37	3.80	2.91	357	316	51.00	985.95	0.80	14.19	3.32
	0.03	272	228	29.46	270.60	0.50	5.97	2.91	322	278	83.40	1371.65	1.47	18.82	3.31
	0.05	320	286	31.07	348.84	0.64	9.97	2.91	264	230	136.88	1794.14	2.81	28.12	3.31
	0.1	354	325	33.01	379.67	0.99	17.50	2.91	225	196	201.26	2333.39	6.00	55.50	3.30
18	0.01	178	163	23.95	210.60	0.30	3.40	3.78	389	374	23.57	657.79	0.30	7.81	4.28
	0.02	206	198	37.42	347.01	0.52	5.50	3.78	354	346	52.63	767.82	0.75	9.72	4.25
	0.03	251	239	39.83	296.28	0.61	4.17	3.78	313	301	93.49	1160.16	1.49	15.74	4.23
	0.05	299	283	42.91	419.34	0.78	8.02	3.78	269	253	156.86	2151.93	3.03	31.50	4.22
	0.1	354	342	39.47	462.37	1.02	11.47	3.78	210	198	210.69	2007.84	5.93	40.50	4.20
20	0.01	142	134	32.09	318.93	0.37	3.68	4.76	420	412	21.35	697.71	0.24	7.64	5.39
	0.02	189	183	42.62	313.54	0.55	5.94	4.76	371	365	60.92	989.82	0.77	12.04	5.35
	0.03	224	220	48.86	374.78	0.69	5.06	4.76	334	330	99.86	1320.37	1.43	19.32	5.34
	0.05	287	283	49.75	459.02	0.85	10.46	4.76	271	267	165.44	1746.21	2.86	29.58	5.31
	0.1	328	327	48.10	616.23	1.19	20.90	4.77	227	226	226.10	2106.65	5.79	45.24	5.28
30	0.01	110	110	82.83	643.77	0.66	5.68	12.03	442	442	22.18	639.43	0.17	5.30	13.17
	0.02	177	177	88.30	644.32	0.78	6.73	12.03	375	375	91.31	1534.67	0.83	13.39	12.99
	0.03	210	210	100.62	666.12	0.99	8.02	12.03	342	342	128.46	1709.04	1.34	16.91	12.91
	0.05	282	282	89.14	899.83	1.11	10.54	12.04	270	270	236.41	2605.58	3.04	30.65	12.82
	0.1	274	273	91.92	693.39	1.64	15.33	12.07	279	278	240.98	2559.28	4.84	48.88	12.77

Table 5
Results of the experiment for the ETI model

n	α	H	TS							SA						
			#	#b	AAD	MAD	ARD [%]	MRD [%]	CPU [seconds]	#	#b	AAD	MAD	ARD [%]	MRD [%]	CPU [seconds]
10	0.01	2	426	145	2.72	84.08	0.06	2.17	1.00	391	110	12.64	331.71	0.29	9.72	1.10
		3	420	144	3.02	88.31	0.07	1.82	1.13	392	116	12.74	352.13	0.27	6.19	1.28
		4	395	167	3.63	81.74	0.08	2.01	1.19	369	141	12.03	287.12	0.26	5.09	1.34
		5	397	168	3.46	103.55	0.08	2.51	1.28	368	139	12.26	353.73	0.26	6.15	1.43
	0.02	2	431	153	4.41	135.04	0.12	4.17	1.00	383	105	23.76	483.24	0.63	17.85	1.15
		3	423	165	5.21	156.53	0.13	3.50	1.13	371	113	22.96	618.62	0.57	11.81	1.28
		4	399	175	6.12	135.01	0.15	3.77	1.19	361	137	24.08	522.72	0.57	9.69	1.33
		5	405	185	5.95	175.42	0.14	4.76	1.28	351	131	25.47	622.17	0.60	11.61	1.43
	0.03	2	436	166	5.29	163.27	0.16	5.98	1.00	370	100	31.87	593.60	0.97	24.73	1.15
		3	429	174	6.86	208.53	0.19	5.14	1.13	362	107	31.60	817.97	0.88	16.93	1.28
		4	411	192	7.14	168.26	0.19	5.31	1.19	344	125	35.40	715.23	0.94	13.85	1.33
		5	421	203	7.63	223.69	0.20	6.74	1.28	333	115	32.14	447.36	0.84	11.62	1.42
	0.05	2	456	193	6.08	178.70	0.23	9.13	1.00	343	80	47.68	1129.25	1.86	35.80	1.15
		3	443	215	8.23	275.80	0.27	8.20	1.13	321	93	48.45	1070.19	1.65	25.83	1.28
		4	432	222	9.25	204.09	0.29	7.75	1.19	314	104	52.12	997.82	1.67	21.09	1.33
		5	440	247	8.83	215.12	0.26	7.91	1.28	289	96	48.56	605.61	1.55	19.47	1.42
	0.1	2	467	240	5.22	162.16	0.31	14.59	1.00	296	69	64.67	941.10	4.24	50.97	1.15
		3	442	258	9.44	271.81	0.47	12.84	1.13	278	94	79.79	1233.84	4.12	42.23	1.28
		4	455	284	10.22	277.02	0.45	14.02	1.19	252	81	86.77	1047.21	4.08	39.91	1.33
		5	458	299	9.82	277.02	0.41	15.34	1.28	237	78	86.89	936.70	3.91	33.36	1.42
12	0.01	2	379	128	5.17	118.91	0.10	2.14	1.42	419	168	13.12	383.30	0.26	8.04	1.63
		3	387	156	4.34	127.56	0.08	2.20	1.59	391	160	15.19	348.95	0.29	7.00	1.79
		4	365	159	4.05	85.18	0.08	2.01	1.66	388	182	16.22	436.70	0.30	8.52	1.86
		5	371	166	4.35	96.69	0.08	2.17	1.79	381	176	14.86	331.55	0.27	5.62	1.98
	0.02	2	390	140	7.58	182.97	0.17	4.06	1.42	407	157	24.42	439.00	0.56	10.86	1.63
		3	389	166	6.87	112.54	0.15	2.76	1.59	381	158	27.64	508.51	0.61	12.80	1.79
		4	369	172	6.79	144.53	0.14	4.06	1.66	375	178	31.81	732.53	0.67	15.29	1.86
		5	374	185	6.90	160.64	0.15	4.36	1.79	362	173	28.50	509.06	0.60	9.33	1.98
	0.03	2	394	170	9.23	163.70	0.24	5.05	1.42	377	153	38.05	566.12	1.01	20.39	1.62
		3	390	188	8.68	157.24	0.21	4.22	1.59	359	157	42.32	605.50	1.08	17.58	1.79
		4	382	190	8.51	165.10	0.20	6.13	1.66	357	165	49.75	906.79	1.17	20.61	1.85
		5	384	208	8.44	183.37	0.20	5.75	1.79	339	163	44.21	641.01	1.03	13.31	1.98
	0.05	2	411	199	10.43	173.62	0.33	7.70	1.42	348	136	54.26	705.47	1.98	29.44	1.62
		3	407	232	9.82	188.32	0.29	6.12	1.59	315	140	62.29	889.60	1.96	24.83	1.79

		4	398	233	9.99	216.45	0.29	10.28	1.66	314	149	75.89	809.15	2.24	28.05	1.85
		5	416	247	9.31	236.80	0.26	9.78	1.79	300	131	71.84	792.76	2.04	20.16	1.97
	0.1	2	435	231	7.91	162.77	0.45	17.19	1.42	316	112	72.50	1263.40	4.60	52.00	1.62
		3	435	279	9.56	200.52	0.46	14.29	1.59	268	112	92.11	1149.96	4.61	46.56	1.79
		4	428	293	9.66	246.69	0.45	19.97	1.66	254	119	102.66	1176.51	4.61	38.91	1.85
		5	437	303	10.13	261.22	0.41	19.45	1.79	244	110	114.21	1230.59	4.90	42.30	1.97
14	0.01	2	286	131	11.04	123.55	0.19	2.18	1.95	420	265	15.35	385.49	0.27	6.71	2.19
		3	316	163	7.79	95.92	0.13	1.48	2.16	388	235	17.21	338.44	0.29	5.56	2.41
		4	295	170	7.89	87.29	0.13	1.43	2.24	381	256	17.46	390.87	0.29	6.11	2.49
		5	279	154	7.95	110.39	0.13	1.75	2.40	397	272	16.43	353.75	0.27	5.66	2.64
	0.02	2	305	146	15.94	220.31	0.32	3.43	1.95	405	246	26.52	488.32	0.56	11.29	2.19
		3	336	196	11.89	161.04	0.23	2.91	2.16	355	215	32.31	538.95	0.62	12.28	2.41
		4	307	194	11.90	118.74	0.22	2.03	2.24	357	244	36.64	591.78	0.69	11.28	2.48
		5	305	195	12.71	183.89	0.24	3.35	2.40	356	246	36.50	658.07	0.68	12.18	2.63
	0.03	2	326	174	17.38	171.30	0.40	4.68	1.95	377	225	40.31	677.37	1.00	15.03	2.18
		3	351	210	14.10	141.83	0.30	4.27	2.17	341	200	52.71	813.47	1.19	15.89	2.40
		4	324	224	14.80	134.03	0.31	2.80	2.24	327	227	55.27	844.26	1.16	14.36	2.48
		5	302	209	15.62	166.12	0.32	3.60	2.40	342	249	52.90	964.57	1.10	19.91	2.63
	0.05	2	358	207	17.79	199.64	0.53	7.32	1.95	344	193	61.04	1457.12	2.01	33.90	2.18
		3	376	259	15.67	209.61	0.42	6.77	2.16	292	175	77.31	992.04	2.26	30.65	2.40
		4	361	275	14.87	161.31	0.37	4.35	2.24	276	190	96.88	1020.29	2.61	25.20	2.47
		5	356	261	16.00	283.56	0.40	7.49	2.40	290	195	90.26	1300.64	2.34	27.56	2.62
	0.1	2	396	259	14.67	175.64	0.82	12.25	1.95	292	155	72.45	1583.54	4.48	51.96	2.18
		3	408	305	13.40	262.48	0.57	11.69	2.17	246	143	105.77	1738.20	5.18	51.53	2.39
		4	404	311	12.50	198.46	0.48	8.61	2.25	240	147	121.97	2318.22	5.17	51.24	2.46
		5	405	331	13.76	272.18	0.51	11.92	2.40	220	146	133.16	1622.03	5.39	50.83	2.61
16	0.01	2	198	98	20.53	149.97	0.31	2.72	2.57	452	352	12.83	404.93	0.20	6.65	2.90
		3	198	124	16.77	145.86	0.24	2.82	2.84	426	352	15.59	399.58	0.22	6.15	3.17
		4	224	163	14.45	146.88	0.21	2.05	2.93	387	326	16.01	450.30	0.23	8.04	3.26
		5	200	152	14.94	147.17	0.21	2.59	3.12	398	350	13.85	310.29	0.20	4.74	3.45
	0.02	2	221	125	29.37	238.74	0.51	5.24	2.57	425	329	25.20	701.26	0.47	13.19	2.89
		3	235	162	24.63	283.15	0.41	4.98	2.84	388	315	37.52	890.10	0.64	13.04	3.15
		4	250	197	22.09	254.40	0.36	4.01	2.93	353	300	35.34	764.12	0.59	12.73	3.24
		5	234	185	23.22	253.27	0.37	5.04	3.12	365	316	32.11	728.10	0.53	11.42	3.43
	0.03	2	255	154	32.18	213.34	0.66	5.38	2.57	396	295	40.85	651.96	0.91	18.26	2.88
		3	263	192	26.52	263.83	0.50	4.77	2.84	358	287	53.54	1130.79	1.04	18.35	3.15
		4	282	225	24.58	331.44	0.44	5.89	2.93	325	268	56.51	1046.30	1.07	19.27	3.23

Table 5 (*continued*)

n	α	H	TS							SA						
			#	#b	AAD	MAD	ARD [%]	MRD [%]	CPU [seconds]	#	#b	AAD	MAD	ARD [%]	MRD [%]	CPU [seconds]
		5	259	224	26.69	344.53	0.47	7.71	3.12	326	291	56.89	867.55	1.04	15.71	3.42
	0.05	2	289	196	31.79	338.26	0.86	7.99	2.57	354	261	66.15	957.90	1.97	33.19	2.88
		3	308	246	25.09	322.07	0.59	8.28	2.84	304	242	87.21	1455.17	2.22	29.40	3.15
		4	331	287	24.59	306.82	0.53	6.59	2.93	263	219	101.88	1544.27	2.40	27.96	3.23
		5	305	270	26.46	513.67	0.57	13.91	3.12	280	245	92.56	1450.54	2.08	26.12	3.42
	0.1	2	327	241	23.17	326.76	1.18	13.79	2.57	309	223	73.97	1398.62	4.40	45.89	2.87
		3	348	305	21.34	312.24	0.87	20.91	2.84	245	202	115.16	2056.18	5.09	63.68	3.14
		4	360	318	25.30	444.81	0.88	16.05	2.93	232	190	125.05	1165.75	4.81	40.99	3.22
		5	349	314	25.13	423.62	0.84	18.43	3.12	236	201	140.11	1305.79	4.92	43.14	3.41
18	0.01	2	131	92	33.69	270.35	0.45	3.46	3.32	460	421	12.18	412.92	0.17	6.06	3.70
		3	172	146	24.92	176.78	0.32	2.02	3.65	406	380	19.18	421.19	0.25	5.06	4.02
		4	179	158	22.06	269.18	0.28	3.04	3.75	394	373	17.00	559.66	0.22	6.36	4.13
		5	166	150	24.70	228.99	0.32	3.27	3.98	402	386	19.16	569.06	0.25	6.39	4.35
	0.02	2	161	120	46.28	411.83	0.73	6.37	3.33	432	391	26.31	730.11	0.44	9.47	3.69
		3	197	168	35.20	224.65	0.52	3.71	3.65	384	355	42.23	807.84	0.65	12.17	4.01
		4	204	187	30.48	413.60	0.44	5.50	3.75	365	348	39.45	888.59	0.60	11.83	4.10
		5	204	188	35.59	381.30	0.52	6.31	3.98	364	348	44.32	1106.61	0.65	13.47	4.34
	0.03	2	201	154	46.45	434.07	0.86	10.49	3.33	398	351	42.38	773.13	0.88	14.55	3.68
		3	235	200	39.01	374.54	0.66	6.17	3.65	352	317	67.99	1071.96	1.21	17.82	3.99
		4	232	204	35.95	420.76	0.59	6.79	3.75	348	320	70.54	1218.50	1.21	17.88	4.09
		5	233	221	35.54	417.45	0.58	8.02	3.98	331	319	73.12	1363.86	1.22	18.86	4.32
	0.05	2	224	179	44.48	350.93	1.13	13.31	3.32	373	328	65.51	1414.66	1.87	26.55	3.67
		3	271	236	37.81	352.19	0.84	8.10	3.65	316	281	92.59	1419.75	2.15	27.52	3.99
		4	274	258	33.88	314.87	0.68	6.62	3.75	294	278	94.18	1490.00	2.07	33.48	4.09
		5	290	275	36.39	308.70	0.73	7.83	3.98	277	262	108.70	1401.50	2.32	29.23	4.31
	0.1	2	278	221	31.23	259.29	1.56	15.94	3.32	331	274	67.04	1059.93	4.14	50.26	3.67
		3	309	280	29.60	368.12	1.15	11.72	3.65	272	243	111.52	1619.25	4.69	50.95	3.98
		4	332	313	30.45	310.45	1.00	12.86	3.75	239	220	135.81	1913.55	4.99	52.13	4.08
		5	333	320	30.43	344.14	0.94	12.39	3.99	232	219	151.33	2065.33	5.17	44.55	4.29
20	0.01	2	87	63	50.30	369.22	0.62	5.36	4.17	491	467	9.52	414.31	0.12	4.81	4.65
		3	143	124	36.85	379.40	0.44	5.25	4.55	430	411	16.22	952.75	0.19	10.49	5.04
		4	137	129	36.16	273.38	0.43	3.25	4.66	425	417	15.46	688.45	0.18	6.78	5.14
		5	136	132	32.12	246.77	0.38	3.12	4.94	422	418	16.04	548.16	0.18	5.91	5.42
	0.02	2	124	96	67.34	411.34	1.00	6.84	4.17	458	430	23.57	634.78	0.37	10.69	4.62

		3	178	161	48.78	363.44	0.67	6.29	4.55	393	376	38.35	1421.48	0.55	19.21	5.01
		4	174	163	49.41	349.71	0.67	5.11	4.66	391	380	43.66	1041.00	0.59	12.30	5.11
		5	182	177	44.38	321.27	0.59	4.68	4.94	377	372	41.25	1066.96	0.55	13.46	5.39
	0.03	2	152	127	70.79	386.60	1.24	8.66	4.17	427	402	35.96	766.17	0.69	15.81	4.61
		3	207	193	50.60	552.65	0.80	10.68	4.55	361	347	59.80	1550.45	1.01	25.72	5.00
		4	200	192	50.99	395.43	0.77	5.89	4.66	362	354	66.43	1001.62	1.06	16.04	5.09
		5	209	200	47.29	324.83	0.70	6.14	4.94	354	345	69.09	1280.71	1.06	18.99	5.37
	0.05	2	191	162	70.58	391.59	1.73	13.56	4.17	392	363	57.45	1216.11	1.56	39.05	4.60
		3	241	230	54.02	413.01	1.11	12.17	4.55	324	313	81.10	1335.47	1.80	30.72	4.99
		4	262	256	46.30	434.81	0.90	9.21	4.66	298	292	119.04	1285.00	2.42	26.79	5.08
		5	252	247	48.63	360.37	0.89	9.55	4.94	307	302	101.46	1224.49	1.98	27.72	5.36
	0.1	2	202	178	48.26	346.70	2.41	22.34	4.17	376	352	56.33	1532.98	3.35	47.60	4.60
		3	292	286	40.48	321.37	1.45	20.02	4.55	268	262	105.15	1430.16	4.39	39.63	4.96
		4	301	294	41.35	363.47	1.30	16.72	4.67	260	253	131.16	1478.41	4.68	34.83	5.07
		5	306	296	41.40	428.38	1.21	15.35	4.94	258	248	143.67	1880.89	4.65	39.46	5.32
30	0.01	2	29	27	153.54	825.39	1.32	7.76	10.49	525	523	2.97	337.72	0.02	2.44	11.33
		3	52	52	128.86	735.51	1.06	6.04	11.28	500	500	8.21	585.38	0.07	5.12	12.09
		4	75	75	106.52	752.29	0.87	6.78	11.45	477	477	12.09	609.08	0.10	4.75	12.26
		5	84	84	92.14	557.69	0.75	5.12	11.99	468	468	11.46	600.54	0.09	4.06	12.81
	0.02	2	42	41	194.12	795.97	2.07	9.57	10.49	511	510	9.06	432.16	0.10	5.73	11.23
		3	90	90	153.29	735.61	1.49	6.93	11.28	462	462	27.90	910.75	0.29	9.26	11.98
		4	122	122	129.67	1132.67	1.23	11.81	11.45	430	430	38.59	1528.89	0.39	12.23	12.14
		5	128	128	113.37	617.04	1.05	7.65	11.99	424	424	39.01	828.86	0.38	8.68	12.67
	0.03	2	68	65	212.09	846.06	2.79	14.88	10.49	487	484	17.06	708.55	0.24	8.31	11.20
		3	112	112	168.94	1184.09	1.97	16.51	11.28	440	440	35.89	998.62	0.44	14.14	11.92
		4	135	135	140.02	914.99	1.56	12.85	11.45	417	417	51.70	1312.13	0.60	15.60	12.05
		5	177	177	113.57	828.08	1.21	9.31	11.99	375	375	83.39	1246.81	0.95	15.70	12.59
	0.05	2	82	79	187.69	812.11	3.66	20.42	10.49	473	470	32.10	1313.20	0.67	22.48	11.17
		3	157	157	144.67	767.07	2.30	15.29	11.28	395	395	80.85	974.20	1.47	22.49	11.85
		4	206	206	114.90	804.48	1.67	11.41	11.46	346	346	113.38	1749.28	1.78	18.58	12.00
		5	228	228	107.98	904.54	1.48	12.15	12.00	324	324	157.07	1496.27	2.34	23.98	12.51
	0.1	2	97	97	125.20	516.28	5.98	36.42	10.49	455	455	23.43	930.90	1.37	37.92	11.18
		3	162	162	123.51	565.34	3.93	20.45	11.29	390	390	67.42	1269.08	2.55	41.20	11.88
		4	211	211	100.44	653.72	2.64	34.04	11.47	341	341	91.24	1151.56	2.80	34.57	11.98
		5	236	236	96.69	659.28	2.27	21.71	12.01	316	316	127.35	2330.14	3.46	46.86	12.48

Table 6
Results of the experiment for the PP model

n	α	T	TS #	TS #b	TS AAD	TS MAD	TS ARD [%]	TS MRD [%]	TS CPU [seconds]	SA #	SA #b	SA AAD	SA MAD	SA ARD [%]	SA MRD [%]	SA CPU [seconds]
10	0.01	3	375	149	4.02	114.24	0.09	2.72	1.62	387	161	10.17	213.32	0.22	4.29	1.73
		4	378	149	3.99	113.20	0.09	2.71	1.43	387	158	9.04	184.19	0.19	4.12	1.55
		6	406	142	3.44	89.22	0.08	2.12	1.24	394	130	8.24	191.43	0.18	4.08	1.37
		12	456	116	1.32	82.95	0.03	2.08	1.05	420	80	6.55	142.38	0.15	3.45	1.19
	0.02	3	383	172	7.42	198.76	0.17	5.19	1.62	364	153	22.57	378.50	0.52	8.63	1.73
		4	391	156	6.73	194.76	0.16	5.15	1.43	380	145	18.26	351.11	0.42	7.84	1.54
		6	413	160	5.91	145.40	0.15	3.94	1.24	376	123	16.45	314.87	0.40	7.73	1.37
		12	466	137	2.27	136.75	0.06	3.98	1.05	399	70	12.81	240.13	0.33	7.07	1.19
	0.03	3	381	199	9.92	260.03	0.25	7.42	1.63	337	155	30.60	458.30	0.78	12.63	1.72
		4	393	172	9.31	252.43	0.24	7.34	1.43	364	143	26.56	480.35	0.68	11.20	1.54
		6	409	165	8.02	182.24	0.22	5.62	1.24	371	127	24.44	712.81	0.64	14.84	1.37
		12	462	143	2.76	170.00	0.09	5.72	1.05	393	74	18.63	323.34	0.56	10.02	1.19
	0.05	3	402	225	13.17	335.60	0.39	11.24	1.63	311	134	45.79	612.11	1.36	19.35	1.72
		4	414	211	11.53	319.15	0.35	11.03	1.44	325	122	39.81	669.47	1.21	15.22	1.54
		6	422	207	10.97	263.29	0.36	8.50	1.24	329	114	37.14	897.34	1.18	21.91	1.36
		12	459	158	3.69	196.37	0.15	8.71	1.05	378	77	24.27	390.56	0.95	14.92	1.19
	0.1	3	430	285	16.08	381.23	0.65	17.79	1.64	251	106	83.92	930.68	3.46	32.93	1.72
		4	433	282	15.09	345.35	0.64	17.17	1.44	254	103	70.06	881.02	3.10	32.31	1.54
		6	436	252	12.21	313.02	0.59	13.36	1.24	284	100	54.70	944.14	2.68	33.23	1.36
		12	476	200	2.34	166.25	0.18	13.94	1.05	336	60	22.77	404.27	1.70	25.59	1.19
12	0.01	3	350	156	4.83	78.00	0.09	2.18	2.35	391	197	13.72	249.34	0.26	5.05	2.47
		4	348	152	4.61	84.71	0.09	1.98	2.07	395	199	13.02	383.92	0.24	7.18	2.21
		6	362	150	4.17	70.61	0.08	2.01	1.78	397	185	12.96	270.35	0.24	4.85	1.95
		12	404	121	3.36	67.82	0.07	2.00	1.51	426	143	10.30	232.76	0.20	4.22	1.70
	0.02	3	365	180	8.24	137.05	0.17	4.33	2.35	367	182	29.54	466.86	0.61	9.31	2.47
		4	362	182	8.00	150.73	0.17	4.03	2.07	365	185	27.00	506.98	0.56	9.09	2.21
		6	374	161	6.68	124.53	0.14	4.05	1.78	386	173	25.38	483.74	0.52	9.56	1.94
		12	403	139	5.20	113.77	0.12	3.98	1.51	408	144	18.56	341.05	0.42	8.46	1.70
	0.03	3	374	200	10.86	193.82	0.25	6.86	2.36	347	173	45.43	675.39	1.03	16.62	2.47
		4	376	198	9.73	201.76	0.22	6.13	2.07	349	171	41.89	723.69	0.96	13.94	2.21
		6	376	182	8.60	165.10	0.20	6.12	1.78	365	171	39.08	628.88	0.92	16.33	1.94
		12	425	163	6.14	143.52	0.17	5.94	1.51	384	122	25.08	427.94	0.66	12.10	1.70
	0.05	3	388	235	13.13	258.86	0.35	11.35	2.36	312	159	66.28	915.96	1.76	20.09	2.46
		4	399	233	11.63	270.13	0.32	10.40	2.08	314	148	63.66	906.75	1.74	20.36	2.20

		6	391	232	10.70	216.45	0.31	10.25	1.79	315	156	54.70	719.73	1.58	19.78	1.94
		12	437	186	5.88	171.86	0.21	9.73	1.51	361	110	36.08	537.25	1.27	21.68	1.69
	0.1	3	412	298	16.11	400.59	0.61	17.02	2.37	249	135	104.17	1340.70	3.84	38.89	2.47
		4	413	291	13.98	327.52	0.56	20.54	2.09	256	134	97.55	1229.68	3.79	36.46	2.21
		6	425	279	9.94	246.69	0.44	19.89	1.79	268	122	77.31	970.12	3.50	40.63	1.94
		12	456	220	4.63	152.56	0.34	18.17	1.51	327	91	32.88	401.11	2.17	23.90	1.69
14	0.01	3	257	148	9.42	100.01	0.15	1.91	3.27	403	294	10.98	408.68	0.17	5.55	3.42
		4	274	167	8.55	99.85	0.14	1.91	2.88	384	277	12.69	322.60	0.21	4.81	3.06
		6	272	160	8.16	128.21	0.14	2.49	2.47	391	279	14.14	375.92	0.23	5.36	2.68
		12	303	107	8.03	80.80	0.14	1.39	2.10	444	248	10.17	353.76	0.17	5.00	2.34
	0.02	3	281	180	15.10	163.27	0.27	2.95	3.27	371	270	30.17	604.66	0.54	11.81	3.41
		4	295	182	14.25	227.47	0.26	4.93	2.88	369	256	30.81	503.14	0.55	8.57	3.05
		6	282	159	14.33	156.69	0.27	3.06	2.47	392	269	26.59	557.96	0.49	9.83	2.66
		12	320	128	11.89	134.92	0.24	2.68	2.10	423	231	21.37	580.11	0.42	9.93	2.32
	0.03	3	288	194	18.89	218.49	0.37	4.38	3.28	357	263	50.98	793.31	1.00	16.64	3.40
		4	314	202	16.24	226.32	0.32	4.14	2.89	349	237	51.82	719.81	1.04	14.30	3.04
		6	305	196	17.36	181.80	0.36	3.47	2.48	355	246	43.59	682.93	0.90	14.57	2.66
		12	348	162	12.56	169.44	0.29	3.87	2.09	389	203	31.52	752.04	0.74	14.98	2.32
	0.05	3	323	247	22.42	252.33	0.52	6.81	3.29	304	228	85.47	1151.58	2.02	24.75	3.39
		4	338	256	19.11	256.78	0.46	7.16	2.90	295	213	80.92	951.52	1.98	23.27	3.03
		6	348	248	17.21	235.38	0.43	6.36	2.48	303	203	64.66	832.31	1.67	21.80	2.65
		12	370	196	13.39	252.14	0.41	7.75	2.10	355	181	44.63	860.89	1.40	22.90	2.31
	0.1	3	375	313	21.00	353.40	0.68	13.21	3.31	238	176	128.98	1821.82	4.42	41.26	3.38
		4	390	322	17.80	369.09	0.62	12.75	2.91	229	161	116.69	1700.13	4.20	40.07	3.03
		6	398	309	14.79	258.17	0.57	16.79	2.49	242	153	100.16	1569.77	4.04	38.79	2.65
		12	413	252	9.28	175.64	0.53	10.63	2.10	299	138	45.66	1247.76	2.73	43.11	2.31
16	0.01	3	180	141	16.23	140.92	0.23	2.43	4.35	409	370	14.27	430.66	0.20	6.85	4.54
		4	211	167	15.21	133.03	0.22	2.32	3.84	383	339	14.64	404.13	0.20	4.76	4.07
		6	223	161	14.93	185.01	0.22	3.24	3.29	389	327	13.03	426.93	0.19	5.13	3.56
		12	181	94	16.74	135.06	0.25	2.01	2.78	456	369	8.74	429.43	0.13	5.10	3.11
	0.02	3	220	178	24.75	190.05	0.38	3.72	4.35	372	330	31.23	662.37	0.48	10.51	4.51
		4	225	182	23.13	181.52	0.37	3.53	3.84	368	325	34.16	757.16	0.54	11.08	4.04
		6	226	176	23.16	172.08	0.38	3.00	3.29	374	324	35.16	642.52	0.57	9.24	3.54
		12	201	122	24.93	197.15	0.43	3.90	2.78	428	349	19.30	634.34	0.33	8.79	3.10
	0.03	3	253	221	29.11	292.75	0.50	6.33	4.36	329	297	63.39	3544.60	1.15	96.95	4.49
		4	259	215	27.88	247.49	0.49	5.42	3.85	335	291	60.31	3483.52	1.13	96.90	4.02
		6	254	204	27.45	341.71	0.50	7.65	3.30	346	296	59.44	3355.24	1.13	96.79	3.52

Table 6 (*continued*)

n	α	T	TS #	TS #b	TS AAD	TS MAD	TS ARD [%]	TS MRD [%]	TS CPU [seconds]	SA #	SA #b	SA AAD	SA MAD	SA ARD [%]	SA MRD [%]	SA CPU [seconds]
		12	232	148	27.19	246.61	0.54	5.65	2.78	402	318	38.37	3026.18	0.83	96.45	3.07
	0.05	3	305	275	32.28	530.30	0.67	13.34	4.38	275	245	96.46	1325.08	1.97	23.51	4.47
		4	304	267	29.77	342.75	0.64	9.35	3.86	283	246	92.02	1378.60	1.92	23.77	4.00
		6	322	280	25.66	408.42	0.57	11.35	3.30	270	228	85.09	1151.14	1.88	26.27	3.51
		12	271	192	27.16	267.16	0.71	7.73	2.78	358	279	47.16	1073.46	1.26	25.72	3.07
	0.1	3	344	317	33.17	602.65	0.99	21.60	4.41	233	206	148.03	1709.40	4.36	43.04	4.47
		4	347	314	31.70	434.69	1.01	20.30	3.88	236	203	127.63	1761.78	3.94	45.73	4.01
		6	362	315	25.05	409.63	0.90	17.89	3.32	235	188	102.56	1123.17	3.64	37.32	3.51
		12	314	231	18.53	189.14	0.92	11.46	2.78	319	236	49.03	971.45	2.58	41.77	3.07
18	0.01	3	156	150	24.16	189.17	0.30	2.71	5.68	402	396	16.78	487.28	0.21	5.48	5.86
		4	172	163	23.12	228.15	0.29	2.85	5.01	389	380	16.70	605.27	0.21	6.28	5.26
		6	158	140	22.73	178.43	0.29	2.77	4.30	412	394	15.43	590.35	0.20	6.18	4.60
		12	148	105	23.63	192.26	0.31	2.43	3.62	447	404	10.78	328.67	0.14	4.71	4.02
	0.02	3	198	186	36.29	308.02	0.50	5.04	5.69	366	354	40.07	1065.29	0.57	12.20	5.82
		4	205	195	33.86	345.94	0.48	5.53	5.02	357	347	40.13	792.84	0.58	9.19	5.21
		6	202	182	33.44	313.87	0.48	4.40	4.31	370	350	32.66	928.77	0.48	11.85	4.56
		12	167	128	34.89	286.91	0.54	5.70	3.63	424	385	26.89	555.88	0.42	8.96	3.99
	0.03	3	224	211	41.66	362.60	0.64	6.48	5.70	341	328	73.72	1230.95	1.13	16.67	5.79
		4	231	216	35.96	363.98	0.56	6.57	5.03	336	321	65.93	976.77	1.05	12.88	5.19
		6	237	217	36.24	354.71	0.58	5.27	4.31	335	315	60.33	1169.02	1.00	16.56	4.54
		12	192	160	37.57	240.25	0.67	4.76	3.62	392	360	41.00	682.47	0.76	12.61	3.98
	0.05	3	288	280	40.54	361.46	0.74	8.14	5.73	272	264	115.94	1673.69	2.19	23.90	5.78
		4	285	276	39.67	382.01	0.75	8.49	5.05	276	267	117.33	2043.52	2.28	29.70	5.18
		6	283	265	35.41	304.69	0.71	8.48	4.32	287	269	95.43	1544.72	1.99	23.59	4.52
		12	240	204	35.55	249.20	0.84	6.98	3.62	348	312	51.69	1143.73	1.30	24.04	3.97
	0.1	3	322	309	42.92	507.87	1.14	12.72	5.79	243	230	158.51	1684.63	4.44	36.74	5.77
		4	318	311	38.17	373.90	1.10	12.56	5.09	241	234	137.39	1569.24	4.12	44.91	5.18
		6	327	306	31.83	353.89	1.00	12.44	4.35	246	225	121.60	1582.45	4.04	34.94	4.53
		12	272	223	27.35	232.51	1.23	11.88	3.63	329	280	53.43	692.52	2.70	35.27	3.95
20	0.01	3	137	134	30.13	301.12	0.35	3.92	7.10	420	417	16.45	577.00	0.19	6.04	7.32
		4	131	124	31.41	251.73	0.36	2.99	6.28	430	423	16.51	513.24	0.18	5.72	6.58
		6	130	124	30.91	275.87	0.36	3.65	5.40	430	424	14.91	627.62	0.17	6.60	5.77
		12	105	81	34.55	243.78	0.42	3.08	4.55	473	449	10.16	417.44	0.12	5.33	5.04
	0.02	3	175	170	42.61	320.36	0.54	5.14	7.12	384	379	48.85	788.18	0.62	10.67	7.27

		4	169	164	44.43	326.64	0.58	5.28	6.29	390	385	37.57	810.62	0.49	11.03	6.52
		6	166	157	43.28	265.40	0.57	5.09	5.41	397	388	33.96	763.70	0.45	9.52	5.73
		12	122	102	49.49	309.13	0.69	6.27	4.55	452	432	21.93	612.23	0.32	8.53	5.02
	0.03	3	231	228	49.49	349.15	0.69	6.53	7.13	326	323	82.62	1010.10	1.19	17.01	7.22
		4	211	204	46.43	422.03	0.66	7.93	6.30	350	343	77.34	1160.45	1.12	15.26	6.50
		6	212	206	47.53	347.38	0.70	6.71	5.41	348	342	68.58	1410.62	1.03	20.43	5.70
		12	171	151	50.45	364.11	0.83	7.52	4.55	403	383	35.65	671.14	0.60	14.22	4.99
	0.05	3	261	258	51.52	578.60	0.88	14.02	7.16	296	293	118.98	1508.72	2.02	23.17	7.18
		4	270	262	48.00	396.86	0.84	9.93	6.33	292	284	114.99	1528.14	2.01	23.91	6.43
		6	258	250	48.78	378.25	0.90	9.97	5.43	304	296	96.09	1834.84	1.79	26.60	5.65
		12	201	180	49.19	405.05	1.07	11.80	4.55	374	353	46.31	783.98	1.09	17.25	4.96
	0.1	3	314	309	49.23	615.17	1.24	25.44	7.25	245	240	168.54	1863.95	4.28	37.52	7.19
		4	317	312	46.35	443.15	1.22	19.02	6.39	242	237	156.23	1597.44	4.22	37.58	6.46
		6	291	286	42.66	538.56	1.25	25.52	5.47	268	263	118.87	1183.35	3.60	33.52	5.67
		12	238	215	36.94	335.27	1.51	21.61	4.56	339	316	53.95	756.88	2.54	31.07	4.95
30	0.01	3	109	109	86.52	608.00	0.68	5.10	17.90	443	443	14.80	402.36	0.12	3.47	17.85
		4	96	96	83.76	630.37	0.66	5.43	15.89	456	456	13.44	590.34	0.10	4.93	16.09
		6	103	103	78.82	598.77	0.63	5.02	13.77	449	449	17.00	550.20	0.13	4.45	14.22
		12	47	47	103.68	721.78	0.85	6.23	11.50	505	505	7.61	386.32	0.06	3.55	12.38
	0.02	3	172	172	93.16	677.05	0.82	7.34	17.97	380	380	63.67	1234.39	0.58	10.43	17.59
		4	158	158	95.24	558.28	0.85	5.41	15.94	394	394	53.12	958.47	0.50	9.47	15.85
		6	157	157	92.71	520.48	0.85	4.86	13.80	395	395	51.82	1003.05	0.48	10.39	14.01
		12	94	94	115.50	577.79	1.12	5.33	11.52	458	458	26.83	633.05	0.27	6.61	12.23
	0.03	3	202	202	108.43	570.05	1.07	7.18	18.03	350	350	108.59	1372.43	1.11	13.26	17.43
		4	206	206	98.57	639.54	0.98	7.18	15.98	346	346	107.87	1531.34	1.14	15.79	15.72
		6	193	193	107.76	594.34	1.12	7.00	13.84	359	359	85.53	1345.78	0.93	15.74	13.89
		12	126	125	127.95	660.85	1.45	8.61	11.53	427	426	42.04	1028.43	0.53	14.55	12.16
	0.05	3	260	260	91.68	650.17	1.11	8.92	18.14	292	292	185.10	2036.30	2.40	24.94	17.29
		4	262	262	87.97	857.87	1.09	10.47	16.07	290	290	182.23	2365.79	2.42	28.34	15.55
		6	256	256	86.06	698.36	1.12	11.68	13.90	296	296	156.52	1705.41	2.23	24.67	13.76
		12	183	183	104.92	562.86	1.60	8.64	11.55	369	369	84.13	1213.59	1.47	20.91	12.03
	0.1	3	252	252	99.83	676.12	1.85	12.64	18.44	300	300	180.20	1852.19	3.56	39.85	17.48
		4	262	262	86.55	658.37	1.69	18.83	16.30	290	290	156.01	1794.36	3.28	33.55	15.73
		6	265	265	79.33	670.30	1.72	14.88	14.04	287	287	134.81	1431.45	3.20	36.96	13.89
		12	169	169	96.91	571.77	2.84	16.77	11.59	383	383	55.92	733.51	2.02	34.60	12.25

go the same search path. This is not the case as far as simulated annealing is concerned since, because of its random character, it is able to find different best solutions for different values of α.

For each payment model considered, the value of α has an important impact on the NPV of the project. If the other parameters of the problem remain fixed (i.e. the number of activities n for the LSP and PAC models, parameters (n, H) and (n, T) for the ETI and PP models, respectively), the value of NPV decreases with the growth of α. This is an immediate observation, when formulae (1)–(4) are taken into account.

In the case of the ETI model, under fixed values of (n, α) the value of NPV grows with the value of parameter H. It is easy to see that the increase of the number of payments has to result in a better net present value of the project.

It is just the other way round for the PP model—the growth of the value of the interval length T results in a worse NPV, which, of course, follows from the fact that in this case the number of payments is decreased.

Let us now comment the results obtained in terms of a comparison between the metaheuristics.

For all the payment models considered, it can be noticed that TS performs better for smaller number of activities, whereas the efficiency of SA grows with the value of n, both in terms of the number of best solutions found and the average and maximal relative deviations. Nevertheless, the maximal relative deviations for SA can be quite large in some cases, which suggests that if the algorithm fails to find a good solution, its result may be quite distant from the one produced by TS. It should be stressed that the maximal deviations for tabu search are, in general, much smaller than for simulated annealing, which suggests that TS finds poor solutions very rarely. It also holds for each payment model that the growth of the discount rate α improves the results produced by tabu search in terms of the number of best solutions found. The relative and maximal deviations from the best known solution increase with the value of α for TS as well as for SA, but for SA they grow more rapidly. This, however, can be explain by the fact that although the value of NPV decreases with the growth of α, the differences in the value of NPV between two different solutions increase with the value of the discount rate. The computational times of both the metaheuristics are very similar for all the models but, generally, they are a fraction shorter for tabu search. It can, however, follow from the number of disc operations made by the algorithms. These operations are made at each iteration—it is obvious that SA visits exactly one solution during one iteration, whereas TS visits several solutions during an iteration and, in consequence, the total number of iterations performed by TS is much smaller.

For the LSP model, it is visible that the number of equal solutions found by both the algorithms is quite large. It follows from the fact mentioned above that, actually, it is a makespan minimization problem and many different feasible solutions generate schedules of identical length independently of the activities' cash flows. The advantage of SA over TS for greater number of activities is most significant in this model, for 30 activities simulated annealing found better solutions for almost all instances. This fact confirms a very high quality of the presented SA algorithm for the MRCPSP problem [32].

The results obtained for the PAC model are similar to these for LSP, although in this case the advantage of SA for larger problems is not that significant. For this payment model, however, both the metaheuristics find different solutions for the same instance very often, especially for greater values of n.

For the regular time interval models—ETI and PP, we can see that the growth of the number of payments H (or shortening the interval length T) improves the performance of TS, whereas SA is better when the payments are made less often. For larger number of activities, although SA outperforms TS, as it has been mentioned before, the growth of H leads to an evident improvement of the TS results, whereas there is a deterioration of the results obtained by SA. This fact confirms the observation that the more the problem approaches its makespan minimization version (by decreasing the number of payments), the better results are produced by SA. If this process reaches the extremum, where there is only one payment at the end of the project, we obtain the LSP model in which SA achieves the most significant advantage.

6. Conclusions

In this paper the multi-mode resource-constrained project scheduling problem with positive discounted cash flows and the maximization of the net present value of all cash flows has been considered. Four common payment models have been examined: lump-sum payment at the completion of the project, payments at activities' completion times, payments at equal time intervals and progress payments. Applications of two local search metaheuristics: simulated annealing and tabu search for the considered problem have been presented. These metaheuristics have been compared on the basis of a computational experiment performed on a set of instances obtained from standard test problems constructed by the ProGen project generator, where, additionally, cash flows were generated randomly with the uniform distribution.

The results of the computational experiment allow to distinguish particular classes of problems, for which it is profitable to use one metaheuristic or the other. Generally, for each payment model the tabu search algorithm performs better for a smaller number of activities, whereas the growth of the number of activities improves the efficiency of the simulated annealing algorithm. On the other hand, under a fixed number of activities, TS gets the advantage over SA with the growth of the discount rate. For the regular time interval models (ETI and PP) TS behaves better when payments are made more often, otherwise SA seems to be more effective. Generally, the more the problem approaches its makespan minimization version (by decreasing the number of payments), the better results are produced by SA. The above features of the proposed metaheuristics can be taken into account when dealing with a particular class of practical MRCPSPDCF problems.

In the future we plan to continue the research on the application of the metaheuristics to different MRCPSPDCF problems. In particular, it will be interesting to add negative cash flows (i.e. cash outflows) to the considered problems. Although we focus on the NPV criterion, we intend to analyze also other financial criteria. In the further research it is planned to construct a branch-and-bound procedure for the considered problems in order to find optimal solutions. Comparing the results of the metaheuristics with optimal solutions will allow to fully evaluate their efficiency.

Acknowledgements

This research has been supported by the Polish State Committee for Scientific Research (KBN) grant no. KBN 4T11F 001 22.

References

[1] E.H.L. Aarts, J.H.M. Korst, Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Wiley, Wiley, Chichester, 1989.
[2] E.H.L. Aarts, J.H.M. Korst, P.J.M. Van Laarhoven, Simulated annealing, in: E.H.L. Aarts, J.K. Lenstra (Eds.), Local Search in Combinatorial Optimization, Wiley, Chichester, 1997, pp. 91–120.
[3] S.M. Baroum, J.H. Patterson, The development of cash flow weight procedures for maximizing the net present value of a project, Journal of Operations Management 14 (1996) 209–227.
[4] S.M. Baroum, J.H. Patterson, An exact solution procedure for maximizing the net present value of cash flows in a network, in: J. Węglarz (Ed.), Project Scheduling Recent Models, Algorithms and Applications, Kluwer Academic Publishers, Dordrecht, 1999, pp. 107–134.
[5] R.B. Bey, R.H. Doersch, J.H. Patterson, The net present value criterion: Its impact on project scheduling, Project Management Quarterly 12 (2) (1981) 35–45.
[6] J. Błazewicz, J.K. Lenstra, A.H.G. Rinnooy Kan, Scheduling subject to resource constraints, Discrete Applied Mathematics 5 (1983) 11–24.

[7] P. Brucker, A. Drexl, R. Möhring, K. Neumann, E. Pesch, Resource-constrained project scheduling: Notation, classification, models and methods, European Journal of Operational Research 112 (1999) 3–41.
[8] V. Cerny, Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm, Journal of Optimization Theory and Applications 45 (1985) 41–51.
[9] R.W. Conway, W.L. Maxwell, L.W. Miller, Theory of Scheduling, Addison-Wesley, Reading, MA, 1967.
[10] N. Dayanand, R. Padman, On modelling payments in projects, Journal of the Operational Research Society 48 (1997) 1123–1130.
[11] N. Dayanand, R. Padman, On payment schedules in contractor client negotiations in projects: An overview of the problem and research issues, in: J. Węglarz (Ed.), Project Scheduling Recent Models, Algorithms and Applications, Kluwer Academic Publishers, Dordrecht, 1999, pp. 477–508.
[12] N. Dayanand, R. Padman, A two stage search heuristic for scheduling payments in projects, Annals of Operations Research 102 (2001) 197–220.
[13] R.H. Doersch, J.H. Patterson, Scheduling a project to maximize its present value: A zero-one programming approach, Management Science 23 (1977) 882–889.
[14] S.E. Elmaghraby, Activity nets: A guided tour through some recent developments, European Journal of Operational Research 64 (1995) 199–215.
[15] S.E. Elmaghraby, W.S. Herroelen, The scheduling of activities to maximize the net present value of projects, European Journal of Operational Research 49 (1990) 35–49.
[16] R. Etgar, Scheduling project activities to maximize the net present value the case of linear time-dependent cash flows, International Journal of Production Research 37 (2) (1999) 329–339.
[17] R. Etgar, A. Shtub, L.J. LeBlanc, Scheduling projects to maximize net present value the case of time-dependent, contingent cash flows, European Journal of Operational Research 96 (1997) 90–96.
[18] S.S. Erengüç, S. Tufekci, C. Zappe, The solution of the time/cost tradeoff problem with discounted cash flows using generalized Bender's decomposition, Naval Research Logistics 40 (1993) 25–50.
[19] F. Glover, Future path for integer programming and links to artificial intelligence, Computers and Operations Research 13 (1986) 533–549.
[20] F. Glover, Tabu search part 1, ORSA Journal of Computing 1 (1989) 190–206.
[21] F. Glover, Tabu search part 2, ORSA Journal of Computing 2 (1990) 4–32.
[22] F. Glover, M. Laguna, Tabu Search, Kluwer Academic Publishers, Norwell, 1997.
[23] R.C. Grinold, The payment scheduling problem, Naval Research Logistics 19 (1972) 123–136.
[24] W.S. Herroelen, E.L. Demeulemeester, B. De Reyck, A classification scheme for project scheduling, in: J. Węglarz (Ed.), Project Scheduling Recent Models, Algorithms and Applications, Kluwer Academic Publishers, Dordrecht, 1999, pp. 1–26.
[25] W.S. Herroelen, E. Gallens, Computational experience with an optimal procedure for the scheduling of activities to maximize the net present value of projects, European Journal of Operational Research 65 (1993) 274–277.
[26] W.S. Herroelen, P. Van Dommelen, E.L. Demeulemeester, Project network models with discounted cash flows: A guided tour through recent developments, European Journal of Operational Research 100 (1997) 97–121.
[27] W.S. Herroelen, P. Van Dommelen, E.L. Demeulemeester, Resource-constrained project scheduling: A survey on recent developments, Computers and Operations Research 25 (1998) 279–302.
[28] O. Icmeli, S.S. Erengüç, A tabu search adaptation to resource constrained project scheduling problem with discounted cash flows, Computers and Operations Research 21 (1994) 841–854.
[29] O. Icmeli, S.S. Erengüç, A branch and bound procedure for the resource constrained project scheduling problem with discounted cash flows, Management Science 42 (1996) 1395–1408.
[30] O. Icmeli, S.S. Erengüç, The resource constrained time/cost tradeoff project scheduling problem with discounted cash flows, Journal of Operations Management 14 (1996) 255–275.
[31] O. Icmeli, S.S. Erengüç, C.J. Zappe, Project scheduling problems: A survey, International Journal of Operations and Production Management 13 (1993) 80–91.
[32] J. Józefowska, M. Mika, R. Różycki, G. Waligóra, J. Węglarz, Simulated annealing for multi-mode resource-contrained project scheduling problem, Annals of Operations Research 102 (2001) 137–155.
[33] J. Józefowska, M. Mika, G. Waligóra, J. Węglarz, Mutli-mode resource-constrained project scheduling problem with discounted positive cash flows—simulated annealing vs. tabu search, Foundations of Computing and Decision Sciences 27 (2) (2002) 97–114.
[34] B. Kazaz, C. Sepil, Project scheduling with discounted cash flows and progress payments, Journal of the Operational Research Society 42 (1996) 1262–1272.
[35] J.E. Kelley, The critical path method: Resource planning and scheduling, in: J.F. Muth, G.L. Thompson (Eds.), Industrial Scheduling, Prentice-Hall, Englewood Cliffs, NJ, 1963, pp. 347–365.
[36] A. Kimms, Maximizing the net present value of a project under resource constraints using a Lagrangian relaxation based heuristic with tight upper bounds, Annals of Operations Research 102 (2001) 221–236.
[37] A. Kimms, Mathematical Programming and Financial Objectives for Scheduling Projects, Kluwer Academic Publishers, Dordrecht, 2001.

[38] S. Kirkpatrick, C.D. Gelatt Jr., M.P. Vecchi, Optimization by simulated annealing, Science 220 (1983) 671–680.

[39] R. Kolisch, Project Scheduling under Resource Constraints Efficient Heuristics for Several Problem Classes, Physica, Heidelberg, 1995.

[40] R. Kolisch, R. Padman, An integrated survey of deterministic project scheduling, OMEGA The International Journal of Management Science 29 (2001) 249–272.

[41] R. Kolisch, A. Sprecher, PSPLIB a project scheduling problem library, European Journal of Operational Research 96 (1997) 205–216.

[42] R. Kolisch, A. Sprecher, A. Drexl, Characterization and generation of a general class of resource-constrained project scheduling problems, Management Science 41 (1995) 1693–1703.

[43] P.J.M. Van Laarhoven, E.H.L. Aarts, Simulated Annealing: Theory and Applications, Reidel, Dordrecht, 1987.

[44] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, E. Teller, Equation of state calculations by fast computing machines, Journal of Chemical Physics 21 (1953) 1087–1092.

[45] K. Neumann, J. Zimmermann, Procedures for resource leveling and net present value problems in project scheduling with general temporal and resource constraints, European Journal of Operational Research 127 (2000) 425–443.

[46] L. Özdamar, On scheduling project activities with variable expenditure rates, IIE Transactions 30 (8) (1998) 695–704.

[47] L. Özdamar, H. Dündar, A flexible heuristic for a multi-mode capital constrained project scheduling problem with probabilistic cash inflows, Computers and Operations Research 24 (12) (1997) 1187–1200.

[48] L. Özdamar, G. Ulusoy, A survey on the resource-constrained project scheduling problem, IIE Transactions 27 (1995) 574–586.

[49] R. Padman, D.E. Smith-Daniels, Early-tardy cost trade-offs in resource constrained projects with cash flows: An optimization-guided heuristic approach, European Journal of Operational Research 64 (1993) 295–311.

[50] R. Padman, D.E. Smith-Daniels, V.L. Smith-Daniels, Heuristic scheduling of resource-constrained projects with cash flows, Naval Research Logistics 44 (1997) 365–381.

[51] J.H. Patterson, A comparison of exact approaches for solving the multiple-constrained resource, project scheduling problem, Management Science 30 (1984) 854–867.

[52] J.P. Pinder, A.S. Marucheck, Using discounted cash flow heuristics to improve project net present value, Journal of Operations Management 14 (1996) 229–240.

[53] B. De Reyck, W.S. Herroelen, An optimal procedure for the resource-constrained project scheduling problem with discounted cash flows and generalized precedence relations, Computers and Operations Research 25 (1998) 1–17.

[54] A.H. Russell, Cash flows in networks, Management Science 16 (1970) 357–373.

[55] R.A. Russell, A comparison of heuristics for scheduling projects with cash flows and resource restrictions, Management Science 32 (1986) 1291–1300.

[56] C. Schwindt, J. Zimmermann, A steepest ascent approach to maximizing the net present value of projects, Mathematical Methods of Operations Research 53 (2001) 435–450.

[57] C. Sepil, Comment on Elmaghraby's and Herroelen's the scheduling of activities to maximize the net present value of projects, European Journal of Operational Research 73 (1994) 185–187.

[58] C. Sepil, N. Ortaç, Performance of the heuristic procedures for constrained projects with progress payments, Journal of the Operational Research Society 48 (1997) 1123–1130.

[59] A. Shtub, R. Etgar, A branch and bound algorithm for scheduling projects to maximize net present value: The case of time dependent, contingent cash flows, International Journal of Production Research 35 (1997) 3367–3378.

[60] J. Skorin-Kapov, Tabu search applied to the quadratic assignment problem, ORSA Journal of Computing 2 (1990) 33–45.

[61] D.E. Smith-Daniels, N.J. Aquilano, Using a late-start resource-constrained project schedule to improve project net present value, Decision Sciences 18 (1987) 617–630.

[62] D.E. Smith-Daniels, R. Padman, V.L. Smith-Daniels, Heuristic scheduling of capital constrained projects, Journal of Operations Management 14 (1996) 241–254.

[63] A. Sprecher, S. Hartmann, A. Drexl, An exact algorithm for project scheduling with multiple modes, OR Spektrum 19 (1997) 195–203.

[64] C.S. Sung, S.K. Lim, A project activity scheduling problem with net present value measure, International Journal of Production Economics 37 (1994) 177–187.

[65] G. Ulusoy, S. Cebelli, An equitable approach to the payment scheduling problem in project management, European Journal of Operational Research 127 (2000) 262–278.

[66] G. Ulusoy, F. Sivrikaya-Şerifoğlu, S. Şahin, Four payment models for the multi-mode resource constrained project scheduling problem with discounted cash flows, Annals of Operations Research 102 (2001) 237–261.

[67] M. Vanhoucke, E.L. Demeulemeester, W.S. Herroelen, Maximizing the net present value of a project with linear time-dependent cash flows, International Journal of Production Research 39 (2001) 3159–3181.

[68] M. Vanhoucke, E.L. Demeulemeester, W.S. Herroelen, On maximizing the net present value of a project under renewable resource constraints, Management Science 47 (2001) 1113–1121.

[69] K.K. Yang, F.B. Talbot, J.H. Patterson, Scheduling a project to maximize its net present value: An integer programming approach, European Journal of Operational Research 64 (1993) 188–198.
[70] K.K. Yang, L.C. Tay, C.C. Sum, A comparison of stochastic scheduling rules for maximizing project net present value, European Journal of Operational Research 85 (1995) 327–339.
[71] D. Zhu, R. Padman, A metaheuristic scheduling procedure for resource-constrained projects with cash flows, Naval Research Logistics 46 (1999) 912–927.