European Journal of Operational Research 198 (2009) 688–696

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

Simulated annealing and tabu search for multi-mode project payment scheduling

Zhengwen He *, Nengmin Wang, Tao Jia, Yu Xu

Department of Industrial Engineering, School of Management, Xi'an Jiaotong University, Xi'an 710049, China

ARTICLE INFO

Article history:
Received 14 May 2006
Accepted 7 October 2008
Available online 17 October 2008

Keywords:
Project scheduling
Progress payments
Net present value
Simulated annealing
Tabu search

ABSTRACT

This paper involves the multi-mode project payment scheduling problem where the activities can be performed with one of several discrete modes and the objective is to assign activities' modes and progress payments so as to maximize the net present value of the contractor under the constraint of project deadline. Using the event-based method the basic model of the problem is constructed and in terms of the different payment rules it is extended as the progress based, expense based, and time based models further. For the strong NP-hardness of the problem which is proven by simplifying it to the deadline subproblem of the discrete time–cost tradeoff problem, we develop two heuristic algorithms, namely simulated annealing and tabu search, to solve the problem. The two heuristic algorithms are compared with the multi-start iterative improvement method as well as random sampling on the basis of a computational experiment performed on a data set constructed by ProGen project generator. The results show that the proposed simulated annealing heuristic algorithm seems to be the most promising algorithm for solving the defined problem especially when the instances become larger. In addition, the effects of several key parameters on the net present value of the contractor are analyzed and some conclusions are given based on the results of the computational experiment.

© 2008 Elsevier B.V. All rights reserved.

1. Introduction

Since the introduction of cash flows in project scheduling problems by Russell (1970), the maximization of the net present value (NPV) has gained increasing attention throughout the literature. This has led to a large amount of models and algorithms presented by Grinold (1972), Doersch and Patterson (1977), Bey et al. (1981), Russell (1986), Smith-Daniels and Aquilano (1987), Smith-Daniels and Smith-Daniels (1987), Elmaghraby and Herroelen (1990), Patterson et al. (1990), Yang et al. (1992, 1995), Herroelen and Gallens (1993), Padman and Smith-Daniels (1993), Özdamar et al. (1998), Icmeli and Erengüc (1994, 1996), Ulusoy and Özdamar (1995), Baroum and Patterson (1996), Burak and Canan (1996), Pinder and Marucheck (1996), Smith-Daniels et al. (1996), Padman et al. (1997), Shtub and Etgar (1997), Zhu and Padman (1997, 1999), De Reyck and Herroelen (1998), Etgar (1999), Neumann and Zimmermann (2000), Kimms (2001), Schwindt and Zimmermann (2001), Ulusoy et al. (2001), Vanhoucke et al. (2001a,b, 2003), Neumann et al. (2003), Mika et al. (2005), Waligóra (2008), and so on.

Project payment scheduling problem which can be considered as a new branch of the max-NPV project scheduling problem involves how to schedule progress payments so as to maximize the NPV of the contractor or/and the client (Herroelen et al., 1997). Considering project payment scheduling problem, several deterministic models are introduced by Dayanand and Padman (1997) to maximize the contractor's NPV. In the models, a deadline is imposed and the number of payments is fixed. Once the total payment to be received from the client is determined, it remains unchanged during the progress of the project. They suggest that the models proposed can be used by both the contractor and the client subject to some modifications. Dayanand and Padman (2001a) present a two-stage procedure in which the first stage consists of a simulated annealing algorithm and in the second stage, activities are rescheduled to improve the NPV. They report that the performance of this approach is significantly better than the problem-dependent heuristics proposed earlier by them. Dayanand and Padman (2001b) also investigate the payment scheduling problems from the client's viewpoint. Several mixed integer linear programming models are introduced according to practical payment rules. The analysis shows that the client obtains the greatest benefit by scheduling the project for early completion such that the payments are not made at regular intervals.

Unlike Dayanand and Padman, Ulusoy and Cebelli (2000) introduce an approach to find an equitable payment schedule of the project from a joint standpoint of the contractor and the client. A double-loop genetic algorithm in which the outer loop represents the client and the inner loop the contractor is proposed to solve

* Corresponding author. Tel.: +86 029 82665643.
E-mail address: zhengwenhe@mail.xjtu.edu.cn (Z. He).

0377-2217/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2008.10.005

for an equitable solution. An example problem is solved and analyzed and some computational results are reported based on the computational experience from a set of 93 problems. Szmereskovsky (2005) supposes that the payment schedule is selected by the client and the contractor protects his interests by selecting the activity schedule to maximize his own NPV and by rejecting the payment schedule if his NPV does not exceed some minimum amount. He constructs a new model which is proved to be NP-hard in the strong sense and develops a branch-and-bound procedure. Results obtained from the new model contradict those of Dayanand and Padman (2001b). He and Xu (2008) analyze the effect of bonus–penalty structure on project payment scheduling and find that the bonus–penalty structure can enhance the flexibility of project payment scheduling and be helpful for the two parties of contract to get more profits from the project synchronously.

For many real-life projects, it is possible to execute each activity in one of several alternative modes which usually represent relations between the consumed resources and the duration of the activity, leading to the well-known discrete time–cost tradeoff problem (DTCTP). DTCTP involves the selection of a set of execution modes in order to achieve a certain objective which has been divided into three parts in the literature. The so-called deadline problem (P_C|T) aims at minimizing the total cost of the project while meeting a given deadline whereas the budget problem (P_T|C) involves minimizing the project duration without exceeding a given budget. A third objective of the problem (P_TC) is to construct the complete and efficient time–cost profile over the set of feasible project durations. DTCTP is a strongly NP-hard optimization problem for general activity networks and has been studied by Crowston and Thompson (1967), Crowston (1970), Robinson (1975), Billstein and Radermacher (1977), Hindelang and Muth (1979), Patterson and Harvey (1979), Elmaghraby and Kamburowski (1992), Erengüc et al. (1993), De et al. (1995, 1997), Demeulemeester et al. (1996, 1998), Skutella (1998), Vanhoucke (2004), Vanhoucke (2005), Akkan et al. (2005), Tareghian and Taheri (2006), Ranjbar et al. (2009), and so on.

Although DTCTP has been combined with the max-npv project scheduling problem by Erengüc et al. (1993), Ulusoy et al. (2001), Vanhoucke (2004), and Mika et al. (2005), to the best of our knowledge, its influence on project payment scheduling has not been studied so far. We call the resulting problem in such a case the multi-mode project payment scheduling problem (MPPSP) where the objective is to assign activities' modes and progress payments so as to maximize the NPV under the constraint of project deadline. In MPPSP if the contractor hopes to improve his/her NPV through advancing payment time he/she has to realize the milestone events to which payments are attached earlier by executing more activities with their crashed modes, causing the cost of the project ascended. In other words, there exists not only the time–cost tradeoff but also the cost–payment tradeoff in this problem, making it worthwhile to be investigated deeply.

The remainder of this paper is organized as follows: In Section 2, the problem is formulated and its models and complexity are presented. Heuristic algorithms for MPPSP are developed in Section 3. Section 4 devotes to the computational experiment. Section 5 concludes the paper.

2. Optimization models

2.1. Problem formulation

We deal with the problem using the event-based method. The project is represented as AoA (Activity-on-Arc) network and cash flows in the project are all attached to events. Consider a project with$N$ activities and M events. Activity n ($n = 1,2,\ldots,N$) can be executed with Q_n modes and its duration and cost under mode q ($q = 1,2,\ldots,Q_n$) are d_{nq} and c_{nq}, respectively. The cost of event m($m = 1, 2, \ldots, M$) is e_m: $e_m = \zeta \cdot \sum_{n\in S_m^1} c_{nq\cdot} + (1-\zeta)\cdot \sum_{n\in S_m^2} c_{nq}$ where S_m^1 is the set of the activities beginning from event m, S_m^2 the set of the activities ending at event m, $\zeta(0 \leqslant \zeta \leqslant 1)$ the distribution proportion of the cost of activity n over its starting event and ending event. The earned value of event m is v_m: $v_m = \sum_{n\in S_m^2} w_n$ where w_n is the earned value of activity n. The number of payments is $K(K \leqslant M)$ and the amount of the k-th ($k = 1, 2, \ldots, K$) payment is denoted as p_k. p_k ($k = 1,2,\ldots,K-1$) equals the product of the contractor's accumulative earned value and the compensation proportion $\theta(0 \leqslant \theta \leqslant 1)$ while the last payment p_K, which must be arranged at the end event of the project, is determined by the following formula: $p_K = U - \sum_{k=1}^{K-1} p_k$ where $U(U = \sum_{n=1}^{N} w_n)$ is the contract price of the project. D and α are the project deadline and the interest rate per period, respectively.

In MPPSP since the cash flows in the project are all connected with events, we need to define three groups of decision variables for the problem as follows:

$$x_{km} = \begin{cases} 1 & \text{if the } k\text{th payment is attached to event } m \\ & k = 1,2,\ldots,K;\ m = 1,2,\ldots,M \\ 0 & \text{otherwise} \end{cases}$$

$$y_{nq} = \begin{cases} 1 & \text{if activity } n \text{ is performed with mode } q \\ & n = 1,2,\ldots,N;\ q = 1,2,\ldots,Q_n \\ 0 & \text{otherwise} \end{cases}$$

$$z_{mt} = \begin{cases} 1 & \text{if event } m \text{ is relaised at period } t \\ & m = 1,2,\ldots,M;\ t = 1,2,\ldots,D \\ 0 & \text{otherwise} \end{cases}$$

Based on the definitions above, three vectors of decision variables can be defined further for the sake of description:

$$\Omega = (x_{km}, m = 1,2,\ldots,M),$$
$$P = (q : y_{nq} = 1, n = 1,2,\ldots,N), \quad \text{and}$$
$$O = (t : z_{mt} = 1, m = 1,2,\ldots,M).$$

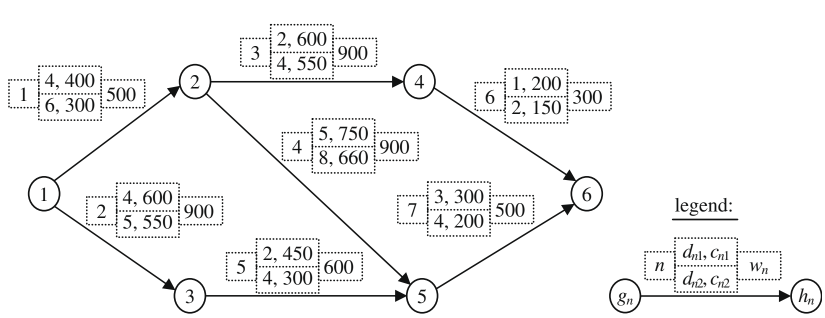

Fig. 1. An example.

It can be seen that given Ω, P, and O, the costs and payments attached to the events can both be determined totally, making the contractor's NPV be able to be worked out.

An example of MPPSP is illustrated as Fig. 1 where all the activities can be performed with two modes. In the figure, the arcs are labeled with the activities' earned values as well as their durations and costs under different modes. Other data of the example are as follows: $\zeta = 0.5$, $\theta = 0.95$, $K = 3$, $\alpha = 0.1$, $D = 14$. Based on the known data, we can calculate U and v_m immediately: $U = w_1 + w_2 + w_3 + w_4 + w_5 + w_6 + w_7 = 4600$, $v_1 = 0$, $v_2 = w_1 = 500$, $v_3 = w_2 = 900$, $v_4 = w_3 = 900$, $v_5 = w_4 + w_5 = 600 + 900 = 1500$, and $v_6 = w_6 + w_7 = 300 + 500 = 800$. Suppose that activity 1 and 4 are performed with mode 1 while the others are executed with mode 2 (i.e. $P = (1,2,2,1,2,2,2)$), then the events' costs can be computed as follows: $e_1 = \zeta \times (c_{11} + c_{22} = 0.5 \times (400 + 550) = 475$, $e_2 = \zeta \times (c_{32} + c_{41}) + (1 - \zeta) \times c_{11} = 0.5 \times (550 + 750) + (1 - 0.5) \times 400 = 850$, $e_3 = \zeta \times c_{52} + (1 - \zeta) \times c_{22} = 0.5 \times 300 + (1 - 0.5) \times 550 = 425$, $e_4 = \zeta \times c_{62} + (1 - \zeta) \times c_{32} = 0.5 \times 150 + (1–0.5) \times 550 = 350$, $e_5 = \zeta \times c_{72} + (1 - \zeta) \times (c_{41} + c_{52}) = 0.5 \times 200 + (1 - 0.5) \times (750 + 300) = 625$, $e_6 = (1 - \zeta) \times (c_{62} + c_{72}) = (1 - 0.5) \times (150 + 200) = 175$. Let us assume further that the three payments are made at event 3, 5, and 6, respectively and all the events are realized at their earliest occurrence times determined by the critical path of the network, leading to that $\Omega = (0,0,1,0,1,0,1)$ and $O = (0,4,5,8,9,13)$. Under the above Ω, P, and O, the project can be finished at 13 which is not later than the deadline, 14, and p_k can be worked out below: $p_1 = \theta \times (v_1 + v_2 + v_3) = 0.95 \times (0 + 500 + 900) = 1330$, $p_2 = \theta \times (v_4 + v_5) = 0.95 \times (900 + 1500) = 2280$, $p_3 = U - (p_1 + p_2) = 4600 - (1330 + 2280) = 990$. Since the cash outflows (i.e. c_m) and the cash inflows (i.e. p_k) are both gotten now, the contractor's NPV can be calculated subsequently: $NPV = (p_1 \times e^{-\alpha\times 5} + p_2 \times e^{-\alpha\times 9} + p_3 \times e^{-\alpha\times 13}) - (c_1 \times e^{-\alpha\times 0} + c_2 \times e^{-\alpha\times 4} + c_3 \times e^{-\alpha\times 5} + c_4 \times e^{-\alpha\times 8} + c_5 \times e^{-\alpha\times 9} + c_6 \times e^{-\alpha\times 13} = 1330 \times e^{-0.1\times 5} + 2280 \times e^{-0.1\times 9} + 990 \times e^{-0.1\times 13}) - (475 \times e^{-0.1\times 0} + 850 \times e^{-0.1\times 4} + 425 \times e^{-0.1\times 5} + 350 \times e^{-0.1\times 8} + 625 \times e^{-0.1\times 9} + 175 \times e^{-0.1\times 13}) = 241.9$.

In order to show the cost-payment tradeoff in MPPSP, now we change the mode of activity 2 from 2 to 1. This variation may result in a new feasible solution: $\Omega = (0,0,1,0,1,0,1)$, $P = (1,1,2,1,2,2,2)$, and $O = (0,4,4,8,9,13)$, and we can calculate that the contractor's NPV under the new solution is 257.8, higher 15.9 than that under the original one. This difference comes from two aspects below. Firstly, the first payment, p_1, which is related to event 3 is advanced from 5 to 4, making the contractor's NPV increased by 84.8: $1330 \times e^{-0.1\times 4} - 1330 \times e^{-0.1\times 5} = 84.8$. Secondly, with the mode variation not only the costs of event 1 and 3, namely e_1 and e_3, ascend but also e_3 occurs earlier, leading to that the contractor's NPV decreases by 68.9: $(500 \times e^{-0.1\times 0} + 450 \times e^{-0.1\times 4}) - (475 \times e^{-0.1\times 0} + 425 \times e^{-0.1\times 5}) = 68.9$. The synthetic effect of the two aspects leads to that the contractor's NPV climbs by 15.9: $84.8 - 68.9 = 15.9$. The fact above means by crashing activities with a cost increment progress payments in the project can be advanced in the meantime, making the contractor's NPV enhanced as a result. Therefore, in MPPSP the contractor can improve the profitability of the project through the tradeoff between the costs and payments.

2.2. Basic model of MPPSP

$$\text{Maximize} \quad \text{NPV} = \sum_{k=1}^{K}\left\{p_k \sum_{m=1}^{M}\left[x_{km}\sum_{t=E_m}^{L_m}(\exp(-\alpha t)z_{mt})\right]\right\} - \sum_{m=1}^{M}\left\{e_m \sum_{t=E_m}^{L_m}[\exp(-\alpha t)z_{mt}]\right\} \tag{1}$$

$$\text{Subject to} \quad \sum_{t=E_m}^{L_m} z_{mt} = 1 \quad m = 1,2,\dots,M \tag{2}$$

$$\sum_{m=1}^{M-1} x_{km} = 1 \quad k = 1,2,\dots,K-1 \tag{3}$$

$$x_{KM} = 1 \tag{4}$$

$$\sum_{k=1}^{K} x_{km} \leqslant 1 \quad m = 1,2,\dots,M \tag{5}$$

$$\sum_{q=1}^{Q_n} y_{nq} = 1 \quad n = 1,2,\dots,N \tag{6}$$

$$\sum_{t=E_{g_n}}^{L_{g_n}} (z_{g_n t}\cdot t) + \sum_{q=1}^{Q_n}(d_{nq}\cdot y_{nq}) \leqslant \sum_{t=E_{h_n}}^{L_{h_n}}(z_{h_n t}\cdot t) \quad n = 1,2,\dots,N \tag{7}$$

$$\sum_{k=1}^{K}\left[p_k \sum_{m=1}^{M}\left(x_{km}\sum_{t=0}^{T} z_{mt}\right)\right] \leqslant \theta \sum_{m=1}^{M}\left[\sum_{t=0}^{T}(v_m z_{mt})\right] \quad T = 1,2,\dots,D \tag{8}$$

$$\sum_{k=1}^{K} p_k = U \quad k = 1,2,\dots,K \tag{9}$$

$$\sum_{t=E_M}^{L_M}(z_{Mt}\cdot t) \leqslant D \tag{10}$$

$$x_{km}, y_{nq}, z_{mt} \in \{0,1\} \tag{11}$$

where g_n and h_n denote the starting event and the ending event of activity n, respectively, E_m, E_{g_n}, and E_{h_n} are the earliest occurrence times and L_m, L_{g_n}, and L_{h_n} the latest occurrence times for event m, g_n, and h_n, respectively.

In the constructed model, the objective function represents the contractor's NPV from undertaking the project and equals the present value of all payments, less that of all costs associated with the project. Constraints (2) arrange an occurrence time for event m within its time window $[E_m, L_m]$; (3) tie payment k ($k = 1, 2,\dots,K - 1$) to a certain event. Constraint (4) makes the last payment K be attached to the end event M. Constraints (5) ensure that at a certain event only one payment can be arranged; (6) choose a mode for each activity in the project; (7) are precedence constraints; (8) are distribution constraints that restrict the total amount of payments by period T to the product of the contractor's accumulative earned value and the compensation proportion. Constraint (9) guarantees that the sum of all payments equals the contract price of the project. Constraint (10) ensures that the occurrence time of the end event of the project is not later than the project deadline. Constraints (11) are the definition fields of the decision variables.

2.3. MPPSP models based on different payment rules

In real-life projects, the contractor and client may come to some special agreements on the arrangement of payment time. Progress based, expense based, and time based payment rules are those adopted frequently in practical contracts. Now we extend the basic model of MPPSP to these cases through changing constraints (3) in it.

2.3.1. Progress based MPPSP model

Generally speaking, the client tends to arrange payment time according to the progress of the project and the contractor is often willing to accept such an arrangement for its fairness. This may lead to the so-called progress based payment rule and according to it the payment time is arranged in proportion to the project progress measured by the contractor's accumulative earned value. By such a payment rule, given a contractor price,

U, and a payment number, K, once the contractor's accumulative earned value reaches to an integer multiples of $[U/K]$ a payment occurs. The progress based payment rule can be formulated formally by

$$x_{km} = \begin{cases} 1 & z_{m\tau} = 1, \\ 0 & \text{otherwise}, \end{cases} \quad k = 1.2, \ldots, K-1; \quad m = 1.2, \ldots, M-1 \tag{12}$$

where τ is determined by the following equation:

$$\tau = \min\left\{T\{z_{mT} = 1\} \cap \left\{k \cdot [U/K] \leqslant \sum_{m=1}^{M}\left[\sum_{t=0}^{T}(v_m z_{mt})\right] \leqslant (k+1) \cdot [U/K]\right\}; T = 1, 2, \ldots, D\right\}.$$

Constraints (12) ensure that payment k (k = 1,2,…,K – 1) is attached to the event which occurs earliest among those making the contractor's accumulative earned value reach $k \cdot [U/K]$ but not surpass $(k+1) \cdot [U/K]$. Replacing constraints (3) in the basic model of MPPSP with constraints (12), we can get the progress based MPPSP model.

2.3.2. Expense based MPPSP model

It is not difficult to understand that the contractor often hopes the payment time may be arranged on the basis of his/her expense spent on the project so that his/her outlay can be compensated duly. Sometimes the contractor's requirement may be approved by the client, resulting in the expense based payment rule. Formally, this payment rule can be expressed as follows: The client makes a payment to the contractor once the latter's expense accumulates to an integer multiples of $[TC/K]$, where TC is the benchmark cost of the project agreed with by the two parties of the contract. In terms of the fact above, the expense based MPPSP model can be constructed by replacing constraints (3) in the basic model with constraints (13)

$$x_{km} = \begin{cases} 1 & z_{m\tau} = 1 \\ 0 & \text{otherwise} \end{cases} \quad k = 1.2, \ldots, K; \quad m = 1.2, \ldots, M-1 \tag{13}$$

where τ satisfies the following equation:

$$\tau = \min\left\{T : \{z_{mT} = 1\} \cap \left\{k \cdot [TC/K] \leqslant \sum_{m=1}^{M}\left[\sum_{t=0}^{T}(c_m z_{mt})\right] \leqslant (k+1) \cdot [TC/K]\right\};\ T = 1, 2, \ldots, D\right\}.$$

Constraints (13) force that the k-th (k = 1,2,…,K–1) payment is made at the occurrence time of the event which is the earliest one among those making the contractor's expense reach$k \cdot [TC/K]$ but not exceed $(k+1) \cdot [TC/K]$.

2.3.3. Time based MPPSP model

In some cases, the payment rule based on the time elapsed from the beginning of the project may be adopted in contracts. This makes payments be arranged in accordance with an equal interval which is determined by the project deadline, D, and the payment number, K. Given D and K, a payment is made whenever the time of integer multiples of $[D/K]$ has passed from the start of the project. Similarly, the time based payment rule can be formulated as constraints (14):

where τ satisfies the following equation:

$$x_{km} = \begin{cases} 1 & z_{m\tau} = 1 \\ 0 & \text{otherwise} \end{cases} \quad k = 1.2, \ldots, K-1; \quad m = 1.2, \ldots, M-1 \tag{14}$$

$$\tau = \min\{T\{z_{mT} = 1\} \cap \{k \cdot [D/K] \leqslant T \leqslant (k+1) \cdot [D/K]\}; T = 1, 2, \ldots, D\}.$$

Constraints (14) guarantee that payment k (k = 1,2,…,K – 1) is tied to the event which is realized earliest within the period of $[k \cdot [D/K], (k+1) \cdot [D/K]]$ and through substituting them for constraints (3) in the basic model we can get the time based MPPSP model.

2.4. Complexity of MPPSP

Without any loss of generality, let K = 1 and α = 0, then the only one payment in MPPSP must be attached to the end event and its amount must equal the contract price. Subsequently, the objective of MPPSP models, i.e. formula (1), may be transformed to maximizing $\text{NPV} = U - \sum_{m=1}^{M} e_m$. For the contract price, U, is a given constant in the problem, this means that in such a condition the objective of MPPSP is equivalent to the minimization of $\sum_{m=1}^{M} e_m$. The fact above indicates that when K = 1 and α = 0 MPPSP is simplified to assigning the modes of activities and the occurrence times of events so as to minimize the total cost of the project under the constraint of the project deadline. Recalling that P_C|T in DTCTP (De et al., 1995) is to identify a realization of activities' modes that minimizes the cost of the project subject to a given deadline, we can conclude immediately that P_C|T is a special case of MPPSP where progress payments and the time value of money are both neglected. In other words, MPPSP can be regarded as a generalization of P_C|T with the consideration of progress payments and the time value of money. Because P_C|T has been proven to be strongly NP-hard for general project networks (De et al., 1997), MPPSP must be strongly NP-hard as well.

3. Heuristic algorithms

In this section the heuristic algorithms, including simulated annealing (SA) and tabu search (TS) which are the well-known metaheuristics that have been successfully applied to a number of project scheduling problems by researchers (e.g. Icmeli and Erengüc, 1994; Boctor, 1996; Shtub et al., 1996; Cho and Kim, 1997; Tsai and Gemmill, 1998; Dayanand and Padman, 2001a; Bouleimen and Lecocq, 2003; Mika et al., 2005; Pan et al., 2008; Lambrechts et al., 2008; Mika et al., 2008; Waligóra, 2008; and so on), are developed for the solution of MPPSP. Besides SA and TS we also present other two methods, namely multi-start iterative improvement (MSII) and random sampling (RS) (Mika et al., 2008; Waligóra, 2008), to provide comparable computational efforts for the metaheuristics.

3.1. Common features

The common elements of the heuristic algorithms are described below.

3.1.1. Solution representation

A feasible solution of MPPSP is represented by three vectors, i.e. Ω, P, and O defined in Section 2.1. In terms of Ω we can determine to which events the payments are attached while the performing modes of activities are defined by P. The occurrence times of events are given by O where the values of the elements must not violate the precedence constraints.

3.1.2. Objective function

As defined in Section 2, the objective function is the contractor's NPV and given a feasible solution, Ω, P, and O, it can be calculated according to the following steps:

[1] Compute the costs of events (i.e. e_m) based on the activities' costs which depend upon their performing modes identified by P.
[2] At each payment event defined by Ω, calculate the contractor's accumulative earned value based on the event occurrence times given by O. Then multiple it by the compensation proportion θ, obtaining the payment amounts, p_k.
[3] Work out the contractor's NPV on the basis of the cash outflows (i.e. e_m), the cash inflows (i.e.p_k), and their occurrence times determined by O.

3.1.3. Starting solution

A starting solution, which is denoted as Ω^0, P^0, and O^0, is generated randomly by the following steps:

[1] Select randomly $K-1$ events from all events except for event M. For Ω, set the values of the elements corresponding to the chosen events and event M at 1 while the others at 0, obtaining Ω^0.
[2] For each activity, select randomly a mode from its available ones, forming a P^0. Compute the time window for each event based on the activities' durations determined by their modes selected. If the earliest occurrence time of the last event is not later than the project deadline accept the P^0 as a feasible one, otherwise, decline it and repeat this step until a feasible P^0 is gotten.
[3] Without the violation of the precedence constraints, arrange an occurrence time for each event within its time window randomly, obtaining a O^0. If the occurrence time of the last event does not exceed the project deadline accept the O^0, otherwise, repeat this step until the gotten O^0 is feasible.

3.2. Simulated annealing

3.2.1. General framework

From the basic MPPSP model it is easy to understand that the arrangement of Ω is independent of that of O and P and vice versa so we divide the simulated annealing heuristic algorithm into two modules, namely Module-PE and Module-ES. The former searches for the desirable Ω on the basis of the given $P$and O whereas the latter seeks the satisfactory P and O under a given Ω. Through the iteration between the two modules the objective can be improved and when its improvement within an iteration is not greater than the operational precision which is denoted as ε and set at 0.1 in this application the algorithm stops and the current solution obtained is regarded as the desirable one. The general framework of the algorithm is illustrated in Fig. 2. Note that the occurrence times of the events can only be assigned after the modes of the activities are determined, so there needs a double-loop searching process in Module-ES. The inner loop searches for the desirable O under the P given by the outer loop while the outer loop determines the desirable P based on the desirable O found in inner loop.

3.2.2. Neighbour generation

During the searching process in Module-PE where P and O keep unchanged, a neighbour of Ω, which is denoted as Ω^1, is generated using the following operator:

– *Element swap* (*ES*): Except for the last element, select randomly an element from the elements whose values are 1 and swap its position with that of another element chosen randomly from those whose values are 0, changing the current Ω to one of its neighbours, Ω^1.Within Module-ES where Ω is given the neighbour solutions of P and O, which are represented as P^1 and O^1, respectively, can be created from their current solutions using the following two operators:
– *Mode changeI* (*MCI*): Select an activity randomly and change its mode determined by the current P to another one arbitrarily. All the modes of activities, including the new mode of this activity as well as the modes of other activities which remain unchanged in the current P, constitute a P^1. Compute the event time windows under the P^1 and judge whether the earliest occurrence time of the last event exceeds the project deadline or not. If the answer is false accept the P^1, otherwise, decline it and repeat this operation until a feasible P^1 is gotten.
– *Time variation* (*TV*): Except for the beginning event, choose an event from all the events randomly. Within the time window of the chosen event, change its occurrence time by a

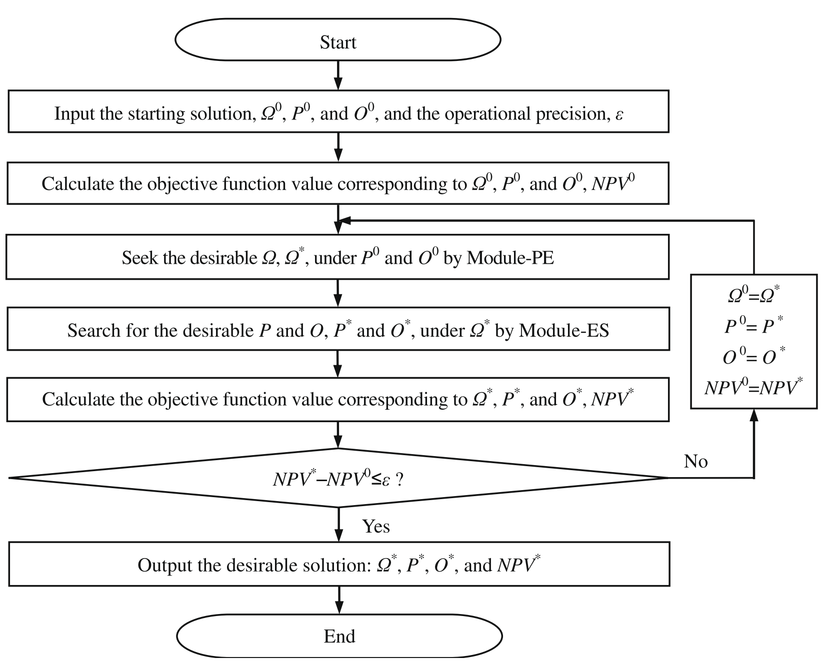

Fig. 2. General frame work of the simulated annealing heuristic algorithm.

unit arbitrarily. If there is the violation of the precedence constraints adjust the occurrence times of other events until the precedence constraints are met, thus obtaining a O^1. Check whether the occurrence time of the last event exceeds the project deadline or not. Accept the O^1 as a feasible one if the answer is false, otherwise, refuse it and repeat this operation until a feasible O^1 is obtained.

3.2.3. Cooling scheme

The cooling scheme of the simulated annealing searching processes in the two modules is described as follows.

- *Initial temperature*: The initial value of the temperature, Temp_0, is calculated from the following equation: $\text{Temp}_0 = \Delta\text{NPV}_{\max}/\ln\text{Prob}_0$, where $\Delta\text{NPV}_{\max}$ is the difference between the maximal objective value and the minimal one which are chosen from the objective values of 50 randomly generated neighbours of the initial solution, and Prob_0, which is set at 0.95 in this application, is the initial acceptance ratio defined as the number of accepted neighbours divided by that of proposed neighbours.
- *Cooling rate*: Beginning from the initial value, Temp_0, the temperature is progressively reduced according to the following equation: $\text{Temp}_{i+1} = \mu \cdot \text{Temp}_i \ (i = 0, 1, \ldots)$ where μ is the cooling rate and it is set at 0.9 in our implementation.
- *Markov chain length*: The length of Markov chains, L, determines the number of transitions for a given value of the temperature. In this implementation, it is calculated according to the following formula: $L = 10 \cdot N$ where N is the number of activities in the project.
- *Stop criterion*: The search process terminates when the current temperature drops to the final temperature which is set at 0.01 in this application.

3.3. Tabu search

3.3.1. Neighbour generation

Operator*ES* and *TV* defined in SA can be used directly to generate neighbour solutions in TS. However, since the time windows of the events may vary with the change of the activities' modes operator*MCI* needs to be modified as follows:

- *Mode changeII* (*MCII*): Generate a feasible neighbour for the current P using *MCI*. Compute the time windows of the events under the neighbour P and check whether the occurrence time of each event is within its time window or not. If the answer is false, without the violation of the precedence constraints adjust the occurrence times of the events into their time windows, changing the current O to a new one with the generation of the neighbour P.

In TS, *ES*, *MCII*, and *TV* are applied independently to the current solution to generate neighbour solutions. For each transition the operators are selected randomly with the probabilities of 0.2, 0.4, and 0.4 for *ES*, *MCII*, and *TV*, respectively.

3.3.2. Moves

Considering the neighbour generation operators utilized in TS, the corresponding moves are defined as follows:

- *Move for ES*: It is a couple of (*position number of the chosen element whose value is 1,position number of the chosen element whose value is 0*). E.g. if element 1 in position #5 is swapped with element 0 in position #2 then the move is represented as $(5,2)$, meaning that a payment is shifted from event 5 to event 2. In the meantime the reverse move which is denoted as (5) is added to the tabu list, forbidding a payment to be reassigned at event 5.
- *Move for MCII*: It is a triple of (*number of the chosen position*, *original value*, *new value*). E.g. if the value in position #3 is changed from 1 to 2 then the move is expressed as $(3,1,2)$, implying that the mode of activity 3 is changed from 1 to 2. Consequently the reverse move, which has the form of $(3,1)$, is added to the tabu list, preventing the mode of activity 3 from being changed back to 1.
- *Move for TV*: It is a triple of (*number of the chosen position,original value*, *new value*). E.g. if the value in position #6 is changed from 13 to12 then the move is described as $(6,13,12)$, indicating that the occurrence time of event 6 is changed from 13 to 12. In consequence, the reverse move can be denoted as $(6,13)$ and it is added to the tabu list, forbidding the assignment of event 6 occurring at 13 again.

3.3.3. Tabu list

The tabu list is managed according to the Tabu Navigation Method (Skorin-Kapov, 1990) and its length in this application is set at 10. Whenever a move performed, its reverse move is added to the tabu list and the oldest existing move is removed from the front of the list according to the First-in-First-out rule. All moves on the tabu list are forbidden. However, if a tabu move can generate a solution better than the best found so far its tabu status may be cancelled in the light of the aspiration criterion so that the algorithm can move to this solution.

3.3.4. Stop criterion

To assure a comparable computational effort for each heuristic algorithm, the total number of the feasible solutions visited by the two modules of SA, which is denoted as Num^{SA}, is recorded and taken as the stop criterion for TS. In other words, when the number of the solutions visited by TS reaches Num^{SA} it terminates and output the best solution saved as the desirable one.

3.4. Multi-start iteration improvement and random sampling

We utilize MSII and RS (Mika et al., 2008; Waligóra, 2008) to provide comparable computational efforts to SA and TS. In our implementation, MSII starts from the same initial solution and employs the same neighbour generation mechanism as those used in TS. The most improving neighbour is chosen and when there are no improving moves it restarts with another random feasible solution. MSII stops and takes the best solution found as the desirable one when the number of the solution it has visited attains Num^{SA}. In RS, Num^{SA} feasible solutions are generated randomly and the best one is selected as the desirable solution.

4. Computational experience

The heuristic algorithms are coded and compiled with Visual Basic 6.0 and the computational experiment is performed on a Pentium-based personal computer with 800 MHz clock-pulse and 128MB RAM. The data set is constructed by ProGen project generator which is available in the project scheduling problem library PSPLIB (Kolisch et al., 1995; Kolisch and Sprecher, 1996). The set consists of 40 instances and the parameter setting used to generate the instances is represented in Table 1. For a given instance the basic model and its three extended versions are solved. Furthermore, in order to analyze the impacts of several key parameters on the contractor's NPV we tackle the each model under the different

Table 1
Parameter setting used to generate the data set.

Number of non-dummy activities	10, 20, 30, or 40
Number of instances generated under a given number of non-dummy activities	10
Number of initial and terminal activities	Randomly selected from 2, 3, and 4.
Maximal number of successors and predecessors	4
Number of performing modes	2
Activity durations with mode 1, d_{n1}	Randomly selected from the interval [1,10]
Activity costs with mode 1, c_{n1}	Randomly selected from the interval [10,20]
Activity durations with mode 2, d_{n2}	$\rho_1 \cdot d_{n1}$, where ρ_1 is randomly selected from the interval [0.8,1]
Activity costs with mode 2, c_{n2}	$\rho_2 \cdot c_{n1}$, where ρ_2 is randomly selected from the interval [1,1.2]
Earned values of activities, w_n	$\rho_3 \cdot c_{n2}$, where ρ_3 is randomly selected from the interval [1.3,1.5]
Project deadline, D	$\rho_5 \cdot (C_{\max} - C_{\min}) + C_{\min}$, where ρ_5 is randomly selected from the interval [0,1], $C_{\max}$ and $C_{\min}$ are the length of critical path of the network when all the activities are performed with modes 1 and 2, respectively

Table 2
Key parameters and their values.

Payment number, K	3, 4, 5
Interest rate per period, α	0.01, 0.02, 0.03
Compensation proportion of the contractor's accumulative earned value, θ	0.75, 0.85, 0.95

values of these parameters which are shown in Table 2. Note that the facts above lead to 108 variants for each instance and 4,320 variants overall.

Four indices are defined to evaluate the performance of the heuristic algorithms:

– *ARD*: Average relative deviation from the best solution known, i.e. the best solution found by any of the heuristic algorithms.
– *MRD*: Maximal relative deviation from the best solution known.
– *ACT*: Average computational time of the heuristic algorithm.
– *MCT*: Maximal computational time of the heuristic algorithm.

The results of the experiment are presented in Table 3 where BM, PBM, EBM, and TBM represent the basic, progress based, expense based, and time based models of MPPSP, respectively.

For the four MPPSP models considered, Table 3 indicates that SA and TS outperform MSII and RS clearly and their superiorities augment with the increase of the number of activities. This result is not surprising because the intelligent search processes generally get an advantage over simple search procedures like MSII and RS and this advantage grows when the problem becomes more complex. Table 3 also shows that the performance of SA is a little better than that of TS and as N increases the *ARD* and *MRD* for SA drop slightly while those for TS climb gently, meaning that SA tends to be more efficient than TS for the larger instances. This result may come from the fact that SA proposed here divides the solution space of the problem into two parts by its two modules, thus enhancing the searching efficiency especially when the instances become larger. As to MSII and RS, it is quite understandable that the former can get better results since it employs the same initial solution and neighbour generation mechanism as those used in TS. RS performs worst from among the heuristic algorithms and it becomes worse rapidly with the increase of the problem scale. This rather proves the fact that the problem is too complicated for a random algorithm to generate good results.

The computational times, which do not exceed 28.81 seconds for all the instances, are ones to be expected – RS is the fastest, then MSII and TS, and SA works the longest. This follows from the fact that RS does not generate neighbors so the times obtained for RS are significantly shorter than those for the others. Contrarily, SA owns the most complex searching process from among the heuristic algorithms, making it become the slowest heuristic algorithm. Concerning TS and MSII, since TS needs to maintain the tabu list additionally compared with MSII it is not strange that it requires more computational efforts to get a satisfactory solution.

Table 3
Results of the experiment.

MPPSP model	N	SA				TS				MSII				RS			
		ARD (%)	MRD (%)	ACT (s)	MCT (s)	ARD (%)	MRD (%)	ACT (s)	MCT (s)	ARD (%)	MRD (%)	ACT (s)	MCT (s)	ARD (%)	MRD (%)	ACT (s)	MCT (s)
BM	10	0.29	1.21	4.93	6.91	0.45	1.22	3.26	4.78	1.68	3.95	2.11	2.35	4.27	6.11	1.53	1.79
	20	0.45	1.20	9.58	12.23	1.16	3.18	7.15	9.53	4.82	9.29	5.03	6.51	9.63	12.45	3.15	4.55
	30	0.22	0.90	16.61	21.26	1.63	3.98	14.64	18.04	8.19	10.9	12.06	14.72	13.10	16.03	9.64	10.67
	40	0.17	0.66	24.80	28.43	2.52	3.74	21.06	24.01	12.48	17.60	17.85	20.19	20.26	24.28	13.92	15.38
PBM	10	0.41	1.12	3.20	3.75	0.11	0.58	2.74	3.29	1.07	2.21	2.08	2.39	3.51	5.00	1.29	1.48
	20	0.27	0.95	6.79	8.81	1.02	2.60	5.87	9.03	4.35	10.00	4.53	6.16	10.08	13.18	2.95	3.88
	30	0.24	0.67	14.06	17.85	2.07	4.60	14.59	16.81	8.70	12.26	12.03	13.97	15.40	19.90	8.14	9.46
	40	0.12	0.27	25.53	28.00	2.03	4.85	22.99	26.11	11.43	15.03	18.74	21.67	19.92	24.39	13.62	15.99
EBM	10	0.33	1.13	3.33	3.73	0.28	0.63	2.70	3.23	1.57	3.02	2.10	2.39	4.16	6.04	1.30	1.47
	20	0.27	0.98	7.89	12.75	0.55	1.74	5.79	8.62	3.75	6.79	4.68	6.52	8.27	10.67	2.83	4.43
	30	0.20	0.86	13.94	17.82	2.04	5.28	14.88	16.78	8.35	12.63	11.71	13.92	14.61	18.21	8.87	9.93
	40	0.10	0.25	25.46	28.50	2.39	5.96	21.72	25.98	12.06	14.41	18.24	22.35	20.04	22.76	13.27	16.03
TBM	10	0.25	1.14	3.65	6.44	0.26	1.06	2.79	3.25	1.14	3.16	2.07	2.33	4.11	5.71	1.41	1.58
	20	0.31	1.20	6.55	8.80	1.17	3.19	5.91	9.43	4.33	8.34	4.60	6.21	9.19	14.37	2.95	4.25
	30	0.28	0.73	14.26	18.11	1.58	3.14	13.82	16.02	9.81	11.57	11.46	14.24	14.47	16.90	7.83	9.37
	40	0.29	0.94	26.36	28.81	2.70	5.67	21.91	24.31	12.88	14.62	17.80	22.27	19.78	22.31	13.13	14.88

Table 4
Effects of the key parameters on contractor's NPV.

Parameters	Values	SA				TS				MSII				RS			
		BM	PBM	EBM	TBM	BM	PBM	EBM	TBM	BM	PBM	EBM	TBM	BM	PBM	EBM	TBM
K	3	98.9	96.5	98.3	96.3	97.8	94.3	94.1	91.69	90.0	86.5	88.6	86.8	89.4	88.4	88.5	87.9
	4	102.9	102.8	103.5	102.5	103.9	101.1	102.2	97.8	96.1	95.5	96.3	94.4	91.0	90.6	90.7	89.6
	5	108.8	107.9	108.0	107.8	107.8	104.3	104.1	102.0	100.8	99.1	99.7	98.7	95.3	91.4	95.1	94.8
α	0.01	144.0	142.7	143.0	142.2	142.5	139.1	140.9	137.2	133.1	130.1	131.8	129.7	125.8	120.8	123.1	123.5
	0.02	105.8	103.3	104.0	102.8	102.8	100.2	100.4	96.8	97.8	90.3	93.6	95.9	90.7	87.0	90.6	89.6
	0.03	76.5	73.5	74.6	73.0	73.9	69.0	70.8	66.9	67.5	68.4	70.2	65.5	66.0	63.8	64.5	62.6
θ	0.75	97.4	94.9	96.0	94.9	94.5	92.0	92.9	93.9	90.0	83.7	86.3	89.9	85.4	82.0	82.6	81.3
	0.85	105.8	103.1	104.3	102.9	102.5	99.6	100.9	98.3	96.6	92.5	92.7	93.4	92.8	89.5	89.9	88.6
	0.95	113.9	110.5	112.5	111.1	111.4	106.6	107.6	106.0	104.7	98.2	100.7	99.6	100.4	97.0	95.3	96.1

The effects of K, α and θ on the contractor's NPV are shown in Table 4 where the average NPVs for the subsets of the instances with the different values of the key parameters are tabulated. From Table 4 it can be seen obviously that the contractor's NPV ascends with the increase of K or θ and descends as α goes up. This can be explained as follows: When K increases the average span between two adjacent payments shortens and thus the contractor's expense can be compensated more quickly, causing the contractor's profit improved. With the increase of θ, the contractor may get more money at each payment during the progress of the project thus his/her NPV rises as well. α reflects the time value of money that is related to the financing cost of the project virtually so when it climbs the contractor's cost for financing the project increases accordingly, making his/her NPVs declined.

Table 4 also shows that the contractor's NPVs under the basic models are not less than those under their corresponding extension models. This is because that the extension models are constructed by replacing constraints (3) in the basic model with constraints (12)–(14), respectively. This in fact reduces the freedom of the payments being arranged on events, making the sets of feasible solutions for the extension models become a kind of subset of feasible solutions for the relative basic models. As a result, the optimal objective values for the extension models cannot exceed those for the corresponding basic models anyway.

5. Conclusions

This paper considers the multi-mode project payment scheduling problem where activities can be performed with one of several discrete modes and the objective is to assign activities' modes and progress payments so as to maximize the contractor's NPV under the constraint of project deadline. Using the event-based method, the basic model of MPPSP is constructed and in terms of the different payment rules it is extended as the progress based, expense based, and time based models further through changing the constraints on payment time. The problem is proven to be strongly NP-hard by simplifying it to P_C|T in the discrete time–cost tradeoff problem (De et al., 1995), and for this attribute owned by the problem the simulated annealing and tabu search heuristic algorithms are developed to solve it. On the basis of a computational experiment performed on a data set constructed by ProGen project generator, the heuristic algorithms are compared with the multi-start iterative improvement method as well as random sampling and the effects of some key parameters on the contractor's NPV are analyzed.

Based on the same total number of the feasible solutions visited and the identical neighbour efficiency, the following conclusion can be drawn from the computational results: The proposed SA seems to be the most promising heuristic algorithm for solving the defined problem, surpassing TS a little while clearly outperforming MSII and RS especially when the instances become larger. The results also show that the contractor's NPV ascends with the increase of the payment number or the compensation proportion whereas descends as the interest rate per period goes up. Another conclusion which can be obtained from the results is that the desirable objective values for the extension models cannot exceed those for the corresponding basic models anyway.

Note that the study in this paper is based on the condition that the activities' modes and the event occurrence times can be arranged without considering resource constraints. However, this is not the case in many practical circumstances where the resource availability for the contractor is limited. Therefore, MPPSP under resource constraints may be a worthwhile problem for further investigation in this direction.

References

Akkan, C., Drexl, A., Kimms, A., 2005. Network decomposition-based benchmark results for the discrete time–cost tradeoff problem. European Journal of Operational Research 165 (2), 339–358.

Baroum, S.M., Patterson, J.H., 1996. The development of cash flow weight procedures for maximizing the net present value of a project. Journal of Operations Management 14, 209–227.

Bey, R., Doersch, R., Patterson, J., 1981. The present value criterion: Its impact on project scheduling. Project Management Quarterly 12 (2), 35–45.

Billstein, N., Radermacher, F.J., 1977. Time–cost optimization. Methods of Operations Research 27, 274–294.

Boctor, F.F., 1996. Resource-constrained project scheduling by simulated annealing. International Journal of Production Research 34 (8), 2335–2351.

Bouleimen, K., Lecocq, H., 2003. A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version. European Journal of Operational Research 149 (2), 268–281.

Burak, K., Canan, S., 1996. Project scheduling with discounted cash flows and progress payments. Journal of the Operational Research Society 47 (10), 1262–1272.

Cho, J.H., Kim, Y.D., 1997. Simulated annealing algorithm for resource constrained project scheduling problems. Journal of the Operational Research Society 48 (7), 736–744.

Crowston, W., 1970. Network reduction and solution. Operations Research Quarterly 21, 435–450.

Crowston, W., Thompson, G.L., 1967. Decision CPM: A method for simultaneous planning, scheduling and control of projects. Operations Research 15, 407–426.

Dayanand, N., Padman, R., 1997. On modeling progress payments in project networks. Journal of the Operational Research Society 48 (9), 906–918.

Dayanand, N., Padman, R., 2001a. A two stage search heuristic for scheduling payments in projects. Annals of Operation Research 102 (1), 197–220.

Dayanand, N., Padman, R., 2001b. Project contracts and payments schedules: The client's problem. Management Science 47 (12), 1654–1667.

De, P., Dunne, E.J., Ghosh, J.B., Wells, C.E., 1995. The discrete time–cost tradeoff problem revisited. European Journal of Operational Research 81, 225–238.

De, P., Dunne, E.J., Ghosh, J.B., Wells, C.E., 1997. Complexity of the discrete time–cost tradeoff problem for project networks. Operations Research 45 (2), 302–306.

Demeulemeester, E., Elmaghraby, S.E., Herroelen, W., 1996. Optimal procedures for the discrete time/cost trade-off problem in project networks. European Journal of Operational Research 88, 50–68.

Demeulemeester, E., De Reyck, B., Foubert, B., Herroelen, W., Vanhoucke, M., 1998. New computational results for the discrete time/cost trade-off problem in project networks. Journal of the Operational Research Society 49, 1153–1163.

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

Doersch, R.H., Patterson, J.H., 1977. Scheduling a project to maximize its present value: A zero-one programming approach. Management Science 23, 882–889.

Elmaghraby, S.E., Herroelen, W.S., 1990. The scheduling of activities to maximize the net present value of projects. European Journal of Operational Research 49, 35–49.

Elmaghraby, S.E., Kamburowski, J., 1992. The analysis of activity networks under generalized precedence relations. Management Science 38, 1245–1263.

Erengüc, S.S., Tufekci, S., Zappe, C.J., 1993. Solving time/cost trade-off problems with discounted cash flows using generalized Benders decomposition. Naval Research Logistics 40, 25–50.

Etgar, R., 1999. Scheduling project activities to maximize the net present value the case of linear time-dependent cash flows. International Journal of Production Research 37 (2), 329–339.

Grinold, R.C., 1972. The payment scheduling problem. Naval Research Logistics Quarterly 19 (1), 123–136.

He, Z., Xu, Y., 2008. Multi-mode project payment scheduling problems with bonus–penalty structure. European Journal of Operational Research 189 (3), 1191–1207.

Herroelen, W., Gallens, E., 1993. 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, 274–277.

Herroelen, W.S., Dommelen, P., Demeulemeester, E.L., 1997. Project network models with discounted cash flows: A guided tour through recent developments. European Journal of Operational Research 100, 97–121.

Hindelang, T.J., Muth, J.F., 1979. A dynamic programming algorithm for decision CPM networks. Operations Research 27, 225–241.

Icmeli, O., Erengüc, S.S., 1994. A tabu search procedure for resource constrained project schedule to improve project scheduling problems with discounted cash flows. Computers and Operations Research 21, 841–853.

Icmeli, O., Erengüc, S.S., 1996. A branch-and-bound procedure for the resource-constrained project scheduling problem with discounted cash flows. Management Science 42, 1395–1408.

Kimms, A., 2001. 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, 221–236.

Kolisch, R., Sprecher, A., 1996. PSPLIB – a project scheduling problem library. European Journal of Operational Research 96, 205–216.

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

Lambrechts, O., Demeulemeester, E., Herroelen, W., 2008. A tabu search procedure for developing robust predictive project schedules. International Journal of Production Economics 111 (2), 493–508.

Mika, M., Waligóra, G., Węglarz, J., 2005. Simulated annealing and tabu search for multi-mode resource-constrained project scheduling with positive discounted cash flows and different payment models. European Journal of Operational Research 164, 639–668.

Mika, M., Waligóra, G., Węglarz, J., 2008. Tabu search for multi-mode resource-constrained project scheduling with schedule-dependent setup times. European Journal of Operational Research 187 (3), 1238–1250.

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

Neumann, K., Schwindt, C., Zimmermann, J., 2003. Order-based neighborhoods for project scheduling with nonregular objective functions. European Journal of Operational Research 149, 325–343.

Özdamar, L., Ulusoy, G., Bayyigit, M., 1998. A heuristic treatment of tardiness and net present value criteria in resource constrained project scheduling. International Journal of Physical Distribution & Logistics Management 28 (9), 805–824.

Padman, R., Smith-Daniels, D., 1993. Early tardy cost trade offs in resource constrained projects with cash flows: An optimization-guided heuristic approach. European Journal of Operational Research 64 (2), 295–311.

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

Pan, N.H., Hsaio, P.W., Chen, K.Y., 2008. A study of project scheduling optimization using Tabu Search algorithm. Engineering Applications of Artificial Intelligence 21 (7), 1101–1112.

Patterson, J.H., Harvey, R.T., 1979. An implicit enumeration algorithm for the time/cost trade-off problem in project network analysis. Foundations of Control Engineering 6, 107–117.

Patterson, J.H., Talbot, F.B., Slowinski, R., Weglarz, J., 1990. Computation experience with a backtracking algorithm for solving a general class of precedence and resource-constrained scheduling problems. European Journal of Operational Research 49, 68–79.

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

Ranjbar, M., De Reyck, B., Kianfar, F., 2009. A hybrid scatter search for the discrete time/resource trade-off problem in project scheduling. European Journal of Operational Research 193, 35–48.

Robinson, D.R., 1975. A dynamic programming solution to cost/time trade-off for CPM. Management Science 22, 158–166.

Russell, A.H., 1970. Cash flows in networks. Management Science 16 (5), 357–373.

Russell, R.A., 1986. A comparison of heuristics for scheduling projects with cash flows and resource restrictions. Management Science 32, 291–300.

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

Shtub, A., Etgar, R., 1997. 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 (12), 3367–3378.

Shtub, A., LeBlanc, L.J., Cai, Z., 1996. Scheduling programs with repetitive projects: A comparison of a simulated annealing, a genetic and a pair-wise swap algorithm. European Journal of Operational Research 88 (1), 124–138.

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

Skutella, M., 1998. Approximation algorithms for the discrete time–cost tradeoff problem. Mathematics of Operations Research 23, 909–929.

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

Smith-Daniels, D.E., Smith-Daniels, V.L., 1987. Maximizing the net present value of a project subject to materials and capital constraints. Journal of Operations Management 7, 33–45.

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

Szmereskovsky, J.G., 2005. The impact of contractor behaviour on the client's payment scheduling problem. Management Science 51, 629–640.

Tareghian, H.R., Taheri, S.H., 2006. On the discrete time, cost and quality trade-off problem. Applied Mathematics and Computation 181 (2), 1305–1312.

Tsai, Y.W., Gemmill, D.D., 1998. Using tabu search to schedule activities of stochastic resource-constrained projects. European Journal of Operational Research 111 (1), 129–141.

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

Ulusoy, G., Özdamar, L., 1995. A heuristic scheduling algorithm for improving the duration and net present value of a project. International Journal of Operations and Production Management 15, 89–98.

Ulusoy, G., Funda, S., Sahin, S., 2001. Four payment models for the multi-mode resource constrained project scheduling problem with discounted cash flows. Annals of Operation Research 102 (1), 237–261.

Vanhoucke, M., 2004. Exact and heuristic procedures for the discrete time/cost trade-off problem under various assumptions. EURO/INFORMS Rhodes Meeting. Greece, 4–7 July.

Vanhoucke, M., 2005. New computational results for the discrete time/cost trade-off problem with time-switch constraints. European Journal of Operational Research 165 (2), 359–374.

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

Vanhoucke, M., Demeulemeester, E., Herroelen, W., 2001b. On maximizing the net present value of a project under renewable resource constraints. Management Science 47, 1113–1121.

Vanhoucke, M., Demeulemeester, E., Herroelen, W., 2003. Progress payments in project scheduling problems. European Journal of Operational Research 148, 604–620.

Waligóra, G., 2008. Discrete–continuous project scheduling with discounted cash flows – a tabu search approach. Computers and Operations Research 35, 2141–2153.

Yang, K.K., Talbot, F.B., Patterson, J.H., 1992. Scheduling a project to maximize its net present value: An integer programming approach. European Journal of Operational Research 64, 188–198.

Yang, K.K., Tay, L.C., Sum, C.C., 1995. A comparison of stochastic scheduling rules for maximizing project net present value. European Journal of Operational Research 85, 327–339.

Zhu, D., Padman, R., 1997. Connectionist approaches for solver selection in constrained project scheduling. Annals of Operations Research 72, 265–298.

Zhu, D., Padman, R., 1999. A metaheuristic scheduling procedure for resource-constrained projects with cash flows. Naval Research Logistics 46, 912–927.