A Scatter Search Algorithm for the RCPSP with Discounted Weighted Earliness-Tardiness Costs

Mohammad Khalilzadeh[1] FereydoonKianfar[2]Mohammad Ranjbar[3]

[1] Department of Industrial Engineering, Sharif University of Technology, Tehran, Iran
khalilzadeh@mehr.sharif.edu
[2] Department of Industrial Engineering, Sharif University of Technology, Tehran, Iran
kianfar@sharif.edu
[3]Department of Industrial Engineering, Faculty of Engineering, Ferdowsi University of Mashhad, Mashhad, Iran
ranjbar@um.ac.ir

Abstract: In this paper, we study a resource-constrained project scheduling problem in which a set of project activities have due dates. If the finish time of each one of these activities is not equal to its due date, an earliness or a tardiness cost exists for each tardy or early period. The objective is to minimize the sum of discounted weighted earliness-tardiness penalty costs of these activities. Scatter Search algorithm is used to deal with this extended form of resource-constrained project scheduling problem. Our implementation of Scatter Search integrates the advanced methods such as dynamic updating of the reference set and the use of frequency-based memory within the diversification generator. Finally, some small and medium size test problems are examined and the computational results are presented. The computational results show the efficiency of the proposed meta-heuristic procedure.
[Mohammad Khalilzadeh, FereydoonKianfar, Mohammad Ranjbar. A Scatter Search Algorithm for RCPSP with Discounted Weighted Earliness-Tardiness Costs. Life Science Journal. 2011;8(2):634-640] (ISSN:1097-8135). http://www.lifesciencesite.com.

Keywords: Earliness-tardiness, net present value, project scheduling, RCPSP, scatter search.

1. Introduction

The resource-constrained project scheduling problem (RCPSP) involves the scheduling of project activities subject to precedence constraints as well as renewable resource restrictions in order to minimize the make span of the project. The RCPSP under minimization of the sum of weighted earliness-tardiness costs (RCPSPWET) is an altered version of the RCPSP in which all assumptions and constraints of the RCPSP are held but the objective has changed. In this paper, we extend the RCPSPWET problem by taking into account the time value of earliness and tardiness costs. We call this problem the RCPSP-DCWET (Resource-Constrained Project Scheduling Problem with Discounted Cash Flows of Weighted Earliness-Tardiness Costs). In the RCPSPWET we assume that a set of project activities have due dates. For each one of these activities if due date is not met, a penalty cost exists for each tardy or early period.

Considerable number of exact and heuristic methods has been presented in the literature for the RCPSP problem with the discounted cash flow, known as RCPSPDC. Russell (Russell, 1986) studied unconstrained resource project scheduling problem with positive and negative cash flows and formulated a non-linear programming model. Elmaghraby and Herroelen (Elmaghraby and Herroelen, 1990) presented an optimal algorithm based on tree structures in activity on arch (AOA) network. Etgar *et al*. (Etgar *et al*., 2003) examined the AOA network of a project scheduling problem assuming that cash flows are associated with events. Shtub and Etgar (Shtub and Etgar, 1997) also offered an exact method to solve the NPV problem with a branch-and-bound approach. Etgar and Shtub again took into account special version of this problem in which cash flows are linear functions of the events realization times. Vanhoucke *et al.* (Vanhoucke *et al.*, 2001c) considered a fixed deadline for the unconstrained max-npv problem.

Some recent studies on the RCPSPDC problems are presented by (Icmeli and Erenguc, 1996), (Smith-Daniels and Aquilano, 1987) and (Vanhoucke *et al.*, 2001a). Doersch, and Patterson (Doersch, and Patterson, 1977) formulated the RCPSPDC with a zero-one integer programming model. Yang et al. Yang *et al.* (Yang *et al.*, 2003) developed a branch and bound method to tackle this problem. Baroum and Patterson (Baroum and Patterson, 1996) developed a branch and bound algorithm for an activity on node (AON) network with non-negative cash flows associated with the activities. Heuristic approaches to the RCPSPDC have been proposed in (Sepil and Ortac, 1997) and (Smith-Daniels and Aquilano, 1987). Some recent surveys on the RCPSPDC are mentioned in (Demeulemeester and Herroelen, 2002). Yang *et al.* (Yang *et al.*, 1995) developed nine stochastic scheduling rules to solve the RCPSPDC problem. Baroum and Patterson (Baroum and Patterson, 1996) introduced a number of priority rule heuristics and discovered their differences based on computational experiments. Pinder and Maruchech (Pinder, and Maruchech, 1996) proposed and compared new scheduling heuristics with different well-known rules. Van houcke (Vanhoucke, 2010) presented a scatter search algorithm for the resource-constrained project scheduling problem with discounted cash flows. He assumed fixed payments associated with the execution of project activities and developed a heuristic optimization procedure to maximize the net present value of the project subject to the precedence and renewable resource constraints.

Another non-regular performance measure, which is gaining attention in just in time environments, is the minimization of the weighted earliness-tardiness penalty costs of the project activities (Demeulemeester and Herroelen, 2002). In this problem setting, activities have an individual activity due date with associated unit earliness and unit tardiness penalty costs. If an activity has been accomplished earlier or later than the predetermined due date, the earliness or tardiness penalty cost can be imposed. The objective then is to schedule the activities in order to minimize the weighted penalty cost of the project subject to the precedence constraints. On the basis of classification scheme introduced by Herroelen *et al.* (Herroelen *et al.*, 1999) the problem can be categorized as cpm|early|tardy. This problem, also known as min-wet problem, is experienced by many firms outsourcing all or some of their activities, such as hiring subcontractors, maintenance crews as well as research teams. Costs of earliness include additional inventory requirements and idle times and implicitly incur opportunity costs. Tardiness may cause customer dissatisfaction or complaints, loss of reputation and profits, monetary penalties and goodwill impairment. Nadjafi and Shadrokh (Nadjafi and Shadrokh, 2009) studied unconstrained resource project scheduling problem considering the time value of the money by continuous discounting the cash flows and minimum as well as maximum time-lags between different activities. They proposed a branch and bound algorithm for this project scheduling problem with generalized precedence relations among activities. The literature on solution procedures for the weighted earliness-tardiness project scheduling problem (WETPSP) is very limited. Vanhoucke (Vanhoucke, 2001) developed an exact recursive search algorithm for unconstrained resource project scheduling problem. The algorithm makes use of the primary idea that each project's earliness-tardiness costs can be minimized by first scheduling activities at their due dates or at a later immediate time if compulsory due to obligatory precedence constraints, followed by a recursive search which figures out the optimal movement for those activities for which a shift towards time zero demonstrates to be favorable. Vanhoucke *et al.* (Vanhoucke *et al.* 2000) used the logic of the recursive approach to solve the WETPSP problem in their branch and bound method for maximizing the net present value of a project in which progress payment takes place. Kazaz and Sepil (Kazaz and Sepil, 1996) solved the WETPSP problem with benders decomposition method. Sepil and Ortac (Sepil and Ortac, 1997) proposed heuristics for the related project scheduling problem under renewable resource constraints. Vanhoucke *et al.* (Vanhoucke *et al.*, 2001b) proposed a branch and bound algorithm to solve the Resource-Constrained Project Scheduling Problem with Weighted Earliness-Tardiness Penalty Costs (RCPSPWET).

In this paper, we extend the RCPSPWET problem by considering the time value of money. This problem is denoted as Resource- Constrained Project Scheduling Problem with Discounted Cash Flows of Weighted Earliness-Tardiness Costs (RCPSP-DCWET). We propose a meta-heuristic-based Scatter Search approach to solve the RCPSP-DCWET in the following sections.

This paper is organized as follows. We commence in Section 2 with the problem modeling and formulation. In Section 3 we describe our schedule representation scheme. In section 4, we briefly review the literature on Scatter Search Algorithm and describe our approach for solving the RCPSP-DCWET problem. Computational results are presented in Section 5. Finally, we end with the conclusions in Section 6.

2. Problem modeling and formulation

The RCPSPWET problem minimizes the weighted earliness-tardiness costs under resource constraints. The project network is depicted by an AON representation where the set of nodes N denotes activities and the set of arcs A indicates finish to start precedence constraints with a zero time lag. Dummy activities 1 and n correspond to start and completion of the project. The list of activities is topologically numbered, i.e., each predecessor of every activity has a smaller number than the number of activity itself.

The parameters of the model are:

D = The set of activities with due date,

e_i = Earliness penalty cost of activity i,

t_i = Tardiness penalty cost of activity i,

d_i= Duration of activity i,

h_i = Due date of activity i,

A= The set finish to start precedence constraints with a zero time lag,

N = The set of activity nodes,
r_{ik}= The resource requirement of activity i for resource type k,
m = The total number of resource types,
a_k = The availability of the k^{th} resource type,
T= The feasible project length,
The variables of the model are:
f_i = The completion time of activity i,
E_i= The earliness of activity i determined by: E_i= max {0, h_i-f_i},
T_i = The tardiness of activity i determined by: T_i = max{0, f_i- h_i},
$S(t)$ = The set of activities that are in progress in time period]t-1,t],
The RCPSPWET can be formulated as follows (Vanhoucke *et al.*, 2001b).

$$\text{Minimize } \sum_{i\in D}(e_i\, E_i + t_i T_i) \qquad (1)$$

Subject to:

$$f_i \le f_j - d_j \qquad \forall (i,j)\in A \qquad (2)$$
$$E_i \ge h_i - f_i \qquad \forall i\in N \qquad (3)$$
$$T_i \ge f_i - h_i \qquad \forall i\in N \qquad (4)$$
$$f_1 = 0 \qquad (5)$$
$$\sum_{i\in S(t)} r_{ik} \le a_k \qquad k = 1,2,\ldots,m, \quad t = 1,2,\ldots,T \qquad (6)$$
$$f_i \ge 0, E_i \ge 0, T_i \ge 0 \text{ and integer} \qquad \forall i\in N \qquad (7)$$

The objective function (1) is to minimize the weighted earliness-tardiness cost of the project, e_i and t_i denote the unit cost of earliness and tardiness for activity i. Equation (2) forces the finish to start precedence constraints among activities. Equation (3) and (4) determine the earliness E_i and tardiness of each activity T_i. Constraint (5) forces the completion time of dummy start activity to be at zero time. Equation (6) introduces the resource constraints. Equation (7) ensures that the activity completion times, earliness and tardiness of activities are non-negative integer values.

Problem Model for the RCPSP-DCWET

In the real world problems, time value of money plays an important role in managerial decision making. Hence, we incorporate Net Present Value (NPV) into the basic form of the RCPSPWET by discounting the cash flow. A continuous discount rate of α is chosen to determine the amount of net present value. Subsequently the continuous discounted factor $e^{-\propto T}$ represents the present value of each unit of money paid at the end of period T, using the discount rate α. Only the objective function of the RCPSP-DCWET is different from the model of the RCPSPWET. The object of this problem is to minimize the net present value of the sum of the earliness-tardiness costs of the activities with due dates, and can be formulated as follows:

$$\text{MinZ}=\sum_{i\in D}\Big(e_i \sum_{k=f_i}^{h_i-1} e^{-\propto k} + t_i \sum_{k=1+h_i}^{f_i} e^{-\propto k}\Big) \qquad (8)$$

subject to constraints (2) to (7) in the RCPSPWET model.

It is easy to show that the RCPSP-DCWET is an extended form of the Resource-constrained Project Scheduling Problem (RCPSP). Since the RCPSP is NP-hard, the RCPSP-DCWET is NP-hard too (Blazewicz *et al.*, 1983).

3. The schedule representation scheme

Knowing the RCPSP-DCWET is NP-hard, we have to abstain from always struggling to solve the corresponding problem instances optimally. Sometimes, the required computation time will just be huge and project managers might effortlessly tend to practical project schedules that are gained within small computation times. This necessity can only be achieved by employing noble heuristic procedures.

We use the serial schedule generation scheme (SSGS) for scheduling activities. This schedule generation scheme (SGS) in line with the parallel schedule generation scheme are two basic ones which have the most efficiency and applications, but usually the serial schedule generation scheme results in better outcomes than the parallel schedule generation scheme. In Lova *et al*. (Lova, *et al.*, 2006) these two schedule generation schemes have been compared in different heuristics and serial schedule generation scheme have generally resulted better. The SSGS adds activities to the schedule until a feasible accomplished schedule is generated. In each it iteration, the next activity in the priority list is chosen and for that activity the first possible starting time is assigned such that no precedence or resource constraint is violated.

We represent a schedule S of the RCPSP-DCWET by a list of activities ($s_1,s_2,s_3,\ldots s_n$) where s_i represents the starting time of activity i. We apply the topological order (TO) condition (Valls, et. al, 2003) by first scheduling the activities using a serial SGS (SSGS) and then sequencing them in non-decreasing order of their finish times, i.e. for all i and j, if $f_i(S) < f_j(S)$, where $f_i(S)$ and $f_j(S)$ indicate the finish time of activities i and j in schedule S, respectively, activity i appears before activity j in the topological ordered activity list. The benefit of this method is that although several activity lists can yield the similar scheduling using a SSGS, each topological order matches a unique schedule, excluding the case of same activity finish times.

4. Our scatter search approach

Scatter Search was first introduced by Glover (Glover, 1977) as a Meta heuristic method that uses a sequence of matched initializations to generate solutions. Scatter Search is an evolutionary population-based algorithm that combines the solutions to obtain new solutions using convex or non-convex linear combinations. The approach of combining existing solutions to generate new ones dates back to 1960s. The intention of this combination mechanism is to integrate both diversity and quality. Recent studies demonstrate the empirical advantages of this meta-heuristic approach for solving a diverse array of optimization problems from both classical and real world settings. We refer the reader to (Marti, 2006) for more information on Scatter Search (SS) algorithm. The general pseudo-code for any Scatter Search method can be outlined as follows:

Algorithm Scatter Search

Diversification Generation Method to produce a pool of various trial solutions, using a random trial solution (or seed solution) as an input.

While Stop Criterion not met:

Improvement Method to convert a trial solution into one or more improved trial solutions (Neither the input nor the output solutions are required to be feasible, though the output solutions will more generally be likely to be so. If no improvement of the input trial solution results, the "improved" solution is considered to be the same as the input solution.)

Reference Set Update Method to construct and uphold a reference set consisting of the b "best" solutions found (where the value of b is usually small, e.g., no more than 20), structured to yield efficient accessing by other parts of the method. Solutions acquire membership to the reference set in accordance with their quality or their diversity.

Subset Generation Method to perform on the reference set, to generate a subset of its solutions as a base for building combined solutions.

Subset Combination Method to transform a particular subset of solutions created by the Subset Generation Method into one or more combined solution vectors.

Scatter Search Illustration

In the following, we describe our Scatter Search algorithm for solving the proposed RCPSP-DCWET problem.

1. Start with P = Ø. Use the diversification generation method to construct a solution and apply the improvement method. Let x be the resulting solution. If $x \notin$ P then add x to P ($P = P \cup x$), otherwise, discard x.

Repeat this step until $|P| = P$ Size.

2. Use the reference set update method to build Ref Set = { x_1, ., x_b } with the best b solutions in P. Order the solutions in Ref Set according to their objective function value such that x_1 is the best solution and x_b the worst.

Make New Solutions = TRUE.

while (New Solutions) do

3. Generate New Subsets with the subset generation method. New Solutions = FALSE.

while (New Subsets ≠ ∅) do

4. Select the next subset s in New Subsets.
5. Apply the solution combination method to s to obtain one or more new trial solution x.
6. Apply the improvement method to the trial solutions.
7. Apply the reference set update method.

if Ref Set has changed then

8. Make New Solutions = TRUE.

end if

9. Delete s from New Subsets.

The above nine-step procedure briefly illustrates the primary framework of our algorithm. At the first stage, an initial population P including $|P|$ solutions is generated. Next, the initial population P is arranged in non-descending order based on their objective functions. In another word, the first solution of P is the best solution (with the lowest objective function) so far. At the third stage, the reference set of high quality solutions, RefSet1 is built. RefSet1 comprises b_1 solutions with the low objective functions. The solutions of RefSet1 are deleted from the list of P initial solutions.

RefSet1=$\{x_1, x_2, x_3, \dots, x_{b1}\}$. The initial population P is updated: $\mathrm{P} = \mathrm{P} - \mathrm{RefSet1}$. At the next stage, the reference set of diverse solutions, RefSet2 is constructed by the following approach: for each initial solution of P, the minimum distance from the RefSet1 solutions is calculated and the initial solution with the maximum distance from the RefSet1 solutions is selected, deleted from P and entered RefSet2:

$$\forall x \in P \; d(x) = \min_{y \in RefSet1} distance(x, y)$$

$$m = \max_{x \in P} d(x) \quad P = P - \{m\}$$

Where $distance(x,y)$ is the Euclidean distance between x and y. This step is repeated b_2 times until the RefSet2 is completed. Hence, RefSet2 contains solutions with high diversity (b=b_1+b_2). At the next stage, the new subsets are produced from r solutions selected from RefSet1 and RefSet2.The number of constructed subsets can be determined by: $\binom{b}{r} = \frac{b!}{r!\,(b-r)!}$.To describe the combination method, we consider the following procedure. Assume we have a precedence feasible solution in which $[i]_l$ represents the activity i located at position $l$and similarly $[j]_u$ denotes the activity j situated at position u. If we swap activities $[i]$ and $[j]$, we obtain a new solution, which may not be precedence feasible. Suppose that activity $[j]$ in position l denoted by $[j]_l$ is precedence feasible and we are seeking for a precedence feasible solution which has activity $[j]$ in position l. For this purpose, we start from position l + 1 and move forward, at each position, say position p, if the activity at position u of the current solution is the precedence of activity x in position p denoted by $[x]_p$, we swap these two activities in position p and u. This move is continued till $p = u$ and we gain the desired precedence feasible solution.

The solutions of each subset are combined to gain preliminary solutions to implement an improvement method. The result of the improvement procedure can trigger the reference set and even the solution population to be updated. Three improvement procedures are employed in our solution method to enhance the efficiency of the algorithm. The first improvement procedure randomly selects a project activity and tries to shift it as the closest as possible to its due date h_i without violating the precedence constraints and the resource restrictions. The second and third improvement procedures are executed simultaneously. The second method combines two solutions selected from the reference set and check the feasibility. Then the first improvement procedure runs automatically. The third improvement method takes into account the earliness or tardiness of all project activities and then finds the maximum value of them and attempts to shift that activity close to its due date considering the precedence constraints.

In the next step, step 6, we exploit the Dynamic Reference Set Updating rule (Ref Set Update Method) as it is likely all the reference solutions are similar and the Scatter Search Methodology will possibly be unable to improve upon the best solution found to execute combinations or improve new trial solutions. The new solution is entered in RefSet1 provided its objective function is better than the worst solution in current Ref Set denoted by x_{b1}. In our minimization problem, if $f(x) < f(x_{b1})$, x is replaced by x_{b1} and RefSet1 is automatically updated. Similarly, the new solution is entered into RefSet2 if its minimum distance from the solutions of RefSet1 is more than the minimum distance of solution in RefSet2 form solutions in RefSet1, $d(x) > d(x_b)$. In another word, while RefSet1 contains the best solutions found so far, RefSet2 is rebuilt from scratch during each iteration.

The main advantage of this method is that the undesireable solutions are taken out from the reference set sooner and consequently the next combined solutions will be better. The combination procedure is narrowed by the reference set which is used as input.

5. Computational Result

In this section, we demonstrate the performance of our proposed Scatter Search algorithm on the problem instances generated by Random Network Generator RanGen (Demeulemeester *et al.*, 2003). Each project test problem has been extended by activity due dates, and unit penalty costs for the earliness and tardiness of the activity completions

For the simplicity of illustration, we suppose the unit earliness costs are equal to the unit tardiness costs. Using fine tuning, we set the size of initial population to $|P| = 10b$, the size of RefSet1, b_1 to $0.75b$ and the size of RefSet2, b_2 to $0.25b$. Consequently for the test problems, the parameters are set as follows: $|P| = 200$, $b = 20$, $b_1 = 15$, $b_2 = 5$, and daily discount rate $\alpha = 0.01(1\%)$.

The Scatter Search procedure has been coded in MATLAB version R2008A under Windows 7 and performed all computational experiments on a laptop (CPU 2.53 GHz processor, and 2 GB of internal memory).

In order to evaluate the performance of the proposed Scatter Search Algorithm, We have generated a problem set of 192 instances using the Random Network Generator RanGen (Demeulemeester *et al.*, 2003). The test problems were generated on the basis of a full factorial design of three parameters, i.e., the number of activities (n), the network shape parameter order strength (OS), and the resource factor (RF). We considered four values 10, 20, 30, and 50 for n, three values 0.25, 0.50 and 0.75 for OS and four values 0.25, 0.50, 0.75 and 1.00 for RF. For each combination of n, OS and RF, we generated four test instances resulting in 4*3*4*4=192 with two resource types. The test problems were extended with unit earliness-tardiness penalty costs for each activity which were randomly selected from the interval 1 and 10. The due dates were generated in the same way as described by Vanhoucke*et al.* (Vanhoucke*et al.,* 2000). We generated random numbers between 1 and maximum due date. The numbers were sorted and assigned to the activities in increasing order. Activity durations were randomly selected from the interval 1 and 10. We also considered the maximum number of 10,000 generated schedules as the termination criterion for our Scatter Search algorithm.

Table.1 represents the average CPU-time and its standard deviation in second for a different number of project activities. Comparing these results with the results obtained by the branch and bound algorithm shown in Table.2, we find out the proposed Scatter Search procedure attains the results close to the optimal in less time comparing with the exact branch and bound algorithm for the problems with 20, 30 and 50 activities. Also the percentage deviations shown in Table.1 are negligible and prove the credibility of the algorithm.

Table.1. The average CPU-time and the standard deviation which Scatter Search algorithm needed to solve RCPSP-DCWET with a different number of activities

Number of activities	Number of Problems	Average CPU-time	Standard Deviation
10	48	0.274	0149
20	48	0.388	0.341
30	48	0.736	0.685
50	48	1.649	0.967

Table.2. The average CPU- time and the standard deviation which Branch and Bound algorithm needed to solve RCPSP-DCWET with a different number of activities

Number of activities	Number of Problems	Average CPU-time	Standard Deviation
10	48	0.002	0.003
20	48	1.476	6.857
30	48	14.389	32.073
50	48	2135.517	4651.966

Table.3. The effect of the Order Strength (OS) for the RCPSP-DCWET

OS factor	Average CPU-time
0.25	1.034
0.50	0.728
0.75	0.523

Table.4. The effect of the Resource Factor (RF) for the RCPSP-DCWET

RF factor	Average CPU-time
0.25	0.531
0.50	0.685
0.75	0.834
1.00	0.997

6. Conclusions

In this paper, we introduced the extended form of the problem of minimizing weighted earliness-tardiness penalty costs in the resource-constrained project scheduling by taking into account the continuous discounted negative cash flows for the first time. Negative cash flows are considered where an activity is accomplished earlier or later than its predetermined due date and negotiated penalty costs may be applied to it. We employed the meta-heuristic-based Scatter Search procedure to tackle this project scheduling problem. The computational results clearly show that the proposed Scatter Search algorithm is effective in solving this kind of combinatorial optimization problem. An interesting research topic that can be examined in the future is developing other meta-heuristic algorithms and benchmarking them for the problem described in this paper.

Corresponding Author:
Mohammad Khalilzadeh
Department of Industrial Engineering, Sharif University of Technology, Tehran, Iran
E-mail: khalilzadeh@mehr.sharif.edu

References

[1]Baroum, S.M. and J.H Patterson, 1996. The Development of Cash Flow Weighted Procedures for Maximizing the Net Present Value of a Project. Journal of Operational Management, 14: 209-227.
[2]J. Blazewicz, J.K. Lenstra, and A.H.G. RinnooyKan, “Scheduling Subject to Resource Constraints: Classification and Complexity”, Discrete Applied Mathematics, Vol. 5, 1983, pp. 11-24.
[3]P. Brucker, A. Drexl, R. Mohring, K. Neumann, and E. Pesch, “Resource-Constrained Project Scheduling: Notation, Classification, Models and Methods”, European Journal of Operational Research, Vol. 112, 1999, pp. 3-41.
[4]Demeulemeester, E. and W. S. Herroelen, 2002 Project Scheduling, A Research Handbook.Kluwer Academic Publishers.
[5]Demeulemeester, E., M. Vanhoucke and W. Herroelen, 2003. A Random Generator for Activity-on-the-node Networks. Journal of Scheduling, 6: 13-34.
[6]Doersch, R.H. and J.H. Patterson, 1977. Scheduling a Project to Maximize its Present Value: A Zero-one Programming Approach. Management Sciences, 23: 882-889.
[7]Elmaghraby, S.E. and W.S. Herroelen, 1990.The Scheduling of Activities to Maximize the Net Present Value of Projects. European Journal of Operational Research, 49: 35-40.
[8]Etgar, R., A. Shtub and L.J. LeBlanc, 1996.Scheduling Projects to Maximize Net Present Value-The Case of Time Independent, Contingent Cash Flows. European Journal of Operational Research, 96: 90-96.

[9]Glover, F., 1977.Heuristics for Integer Programming Using Surrogate Constraints. Decision Sciences, 8: 156-166.
[10]Herroelen, W., B. De Reyck and E. Demeulemeester, 1999.A Classification Scheme for Project Scheduling Problems, Chapter 1. In: Handbook of Recent Advances in Project Scheduling, Weglarz, J. (Ed.). Kluwer Academic Publishers, Dordrecht.
[11]Icmeli, O. and S.S. Erenguc, 1996. A Branch and Bound Procedure for Resource Constrained Project Scheduling Problem with Discounted Cash Flows. Management Sciences, 42: 1395-1408.
[12]Kazaz, B. and C.B. Sepil, 1996.Project Scheduling with Discounted Cash Flows and Progress Payments. Journal of the Operational Research Society, 47: 1262-1272.
[13]Lova, A., P. Tormos, and F. Barber, 2006. Multimode resource-constrained project scheduling: scheduling schemes, priority rules and mode selection rules. In teligenica Artificial, 10: 69-86.
[14]Marti, R., M. Laguna and F. Glover, 2006.Principles of Scatter Search. European Journal of Operational Research, 169: 359-372.
[15]Nadjafi, B.A. and Sh. Shadrokh, 2009. An Algorithm for the Weighted Earliness-Tardiness Unconstrained Project Scheduling Problem. Journal of Applied Sciences, 8: 1651-1659.
[16]Pinder, J.P. and A.S. Maruchech, 1996. Using Discounted Cash Flow Heuristics to Improve Project Net Present Value, Journal of Operations Management, 14: 229-240.
[17]Russell, R.A., 1986. A Comparison of Heuristics for Scheduling Projects with Cash Flows and Resource Restriction. Management Science, 32: 1291-1300.
[18]Sepil, C. and N. Ortac, 1997.Performance of the Heuristic Procedures for Constrained Projects with Progress Payments. Journal of the Operational Research Society, 48: 1123-1130.
[19]Shtub, A. and R. Etgar, 1997. A branch and Bound Algorithm for Scheduling Projects to Maximize Net Present Value: The Case of Time Dependent, Continuous Cash Flows. International Journal of Production Research, 35: 3367-3378.
[20]Smith-Daniels, D.E and N.J. Aquilano, 1987.Using a Late-Start Resource-Constrained Project Schedule to Improve Project Net Present Value. Decision Sciences, 18: 369-387.
[21]Valls, V., Quimtanilla, M.S., Ballestin, F., 2003. Resource-constrained project scheduling: A critical reordering heuristic. European Journal of Operational Research,165: 375-386.
[22]Vanhoucke, M., E. Demeulemeester and W. Herroelen, 2000.Maximizing the Net Present Value of a Project with Progress Payments. Research Report 0028, Department of Applied Economics, KatholiekeUniversiteit Leuven.
[23]Vanhoucke, M., 2001.Exact Algorithm for Various Types of Project Scheduling Problems Nonregular Objectives and Time/cost Trade-offs. Ph.D. Thesis, KatholiekeUniversiteit Leuven.
[24]Vanhoucke, M, E. Demeulemeester and W. Herroelen, 2001a. On Maximizing the Net Present Value of a Project Under Renewable Resource Constraints. Management Science, 48: 1113-1121.
[25]Vanhoucke, M., E. Demeulemeester and W. Herroelen, 2001b. An Exact Procedure for the Resource Constrained Weighted Earliness-Tardiness Project Scheduling Problem. Annals of Operations Research, 102: 179-196.
[26]Vanhoucke, M., E. Demeulemeester and W. Herroelen, 2001c.Scheduling Projects with Linearly Time-dependent Cash Flows to Maximize the Net Present Value. International Journal of Production Research, 39: 3159-3181.
[27]Vanhoucke, M., 2010. A scatter search heuristic for maximizing the net present value of a resource-constrained project with fixed activity cash flows. International Journal of Production Research, 48: 1983-2001.
[28]Yang k.k., F.B. Talbot and J.H. Patterson, 1993.Scheduling a Project to Maximize its Net Present Value: An Integer Programming Approach. European Journal of Operational Research, 64: 188-198.
[29]Yang, K.K., L.C. Tay and C.C. Sum, 1995.A Comparison of Stochastic Scheduling Rules for Maximizing Project Net Present Value. European Journal of Operational Research, 85: 327-339.

5/2/2011