Journal of Industrial Engineering 1 (2008) 35 - 39

The preemptive resource-constrained project scheduling problem subject to due dates and preemption penalties: An integer programming approach

B. Afshar Nadjafi
Department of Industrial Engineering, Islamic Azad University of Qazvin, Qazvin, Iran,
Corresponding Author, Afsharnb@Mehr.sharif.edu

S. Shadrokh
Department of Industrial Engineering, Sharif University of Technology, Tehran, Iran,
Shadrokh@Sharif.edu

Abstract
Extensive research has been devoted to resource constrained project scheduling problem. However, little attention has been paid to problems where a certain time penalty must be incurred if activity preemption is allowed. In this paper, we consider the project scheduling problem of minimizing the total cost subject to resource constraints, earliness-tardiness penalties and preemption penalties, where each time an activity is started after being preempted; a constant setup penalty is incurred. We propose a solution method based on a pure integer formulation for the problem. Finally, some test problems are solved with LINGO version 8 and computational results are reported.

Keywords*:* Project scheduling; Resource constrained; Preemptive scheduling; Earliness-tardiness cost.

1. Introduction

Preemptive project scheduling problems are those in which the accomplishing of an activity can be temporarily interrupted, and restarted at a later time. Consequently in the literature on preemptive project scheduling, preempted activities can simply be resumed from the point at which preemption occurred at no cost. However, this situation is not always true in practice. It is likely that in some cases, a certain delay or setup cost must be incurred.

The literature on solution methods for the *p*reemptive *r*esource *c*onstrained *p*roject *s*cheduling *p*roblem with *w*eighted *e*arliness *t*ardiness and *p*reemption *p*enalties (PRCPSPWETPP) is scant. Of course, several papers have been devoted to machine scheduling with preemption penalties. Potts and Van Wassenhove (Potts & Van Wassenhove, 1992, 395-406) suggested considering preemption penalties under the lot-sizing model. Then, Monma and Potts (Monma & Potts, 1993, 981-993) and Chen (Chen, 1993, 1303-1318) studied the preemptive parallel machine scheduling problem with batch setup times. Zdrzalka (Zdrzalka, 1994, 60-71), Schuurman and Woeginger (Schuurman & Woeginger, 1999, 759-767) and Liu and Cheng (Liu & Cheng, 2002, 107-111) studied preemptive scheduling problems with job dependent setup times. Julien and et al. (Julien & et al, 1997, 359-372) proposed more preemption models and applied them to two single machine scheduling problems.

In project scheduling field, Vanhoucke (Vanhoucke, 2001) and Vanhoucke and et al. (Vanhoucke & et al, 2000a, 179-196) have developed an exact recursive search algorithm for the basic form of *w*eighted *e*arliness-*t*ardiness *p*roject *s*cheduling *p*roblem (WETPSP) in the absence of resource constraints and preemption. The algorithm exploits the basic idea that the earliness-tardiness costs of a project can be minimized by first scheduling activities at their due date or at a later time instant if forced so by binding precedence constraints, followed by a recursive search which computes the optimal displacement for those activities for which a shift towards time zero proves to be beneficial. Vanhoucke and et al. (Vanhoucke & et al, 2000b) have exploited the logic of the recursive procedure for solving the WETPSP in their branch and bound procedure for maximizing the net present value of a project in which progress payments occur. Kaplan (Kaplan, 1988) was the first to study the *p*reemptive *r*esource-*c*onstrained *p*roject *s*cheduling *p*roblem (PRCPSP). She formulated the PRCPSP as a dynamic program and solved it using a reaching procedure. Demeulemeester and Herroelen (Demeulemeester & Herroelen, 1996, 334-348) developed a branch and bound algorithm for the problem.

In this paper, we consider the project scheduling problem of minimizing the total cost subject to resource constraints, earliness-tardiness penalties and preemption penalties, where each time an activity is started after being preempted; a constant setup penalty is incurred. The paper is organized as follows: Section 2 describes the problem. An integer formulation is given in section 3. A numerical example and computational results are represented in section 4 and 5, respectively. Section 6 contains the conclusions.

2. Problem description

A non-regular performance measure, which is gaining attention in just-in-time environments, is the minimization individual activity due date with associated unit earliness and unit tardiness penalty costs.

In the classical resource-constrained project scheduling problem (RCPSP) there is no room for preemption of the activities in the project. Preemptive project scheduling problems are those in which the accomplishing of an activity can be temporarily interrupted, and restarted at a later time. Consequently in the literature on preemptive project scheduling, preempted activities can simply be resumed from the point at which preemption occurred at no cost. However, this situation is not always true in practice. It is likely that in some cases, a certain delay or setup cost must be incurred.

The deterministic preemptive resource-constrained project scheduling problem with weighted earliness tardiness and preemption penalties (PRCPSPWETPP) involves the scheduling of project activities in order to minimize the total earliness-tardiness and preemption penalties of the project in the presence of resource constraints.

In sequent, assume a project represented in *AON* format by a directed graph $G = \{N, A\}$ where the set of nodes, *N*, represents activities and the set of arcs, *A*, represents finish-start precedence constraints with a time-lag of zero. The preemptable activities are numbered from the dummy start activity 1 to the dummy end activity *n* and are topologically ordered, i.e. each successor of an activity has a larger activity number than the activity itself. The fixed duration of an activity *i* is denoted by d_i $(1 \le i \le n)$, while h_i denotes its deterministic due date.

The objective of the PRCPSPWETPP is to schedule a number of activities, in order to minimize the total cost of the project subject to finish to start precedence relations with a time-lag of zero, constrained resources and a fixed deadline.

3. Problem formulation

We have the following notations for preemptive resource-constrained project scheduling problem with weighted earliness tardiness and preemption penalties (PRCPSPWETPP):

n :	number of activities
A :	set of arcs of acyclic digraph representing the project
N :	set of nodes of acyclic digraph representing the project
d_i :	duration of activity i
h_i :	due date of activity i
EST_i :	earliest start time of activity i
LST_i :	latest start time of activity i
EFT_i :	earliest finish time of activity i
LFT_i :	latest finish time of activity i
a_k :	availability of the kth resource type
$r_{:i}$	resource requirement of activity i for resource
T :	deadline of the project
Z :	objective function
π_i :	each preemption penalty of activity i
ν_i :	per unit earliness cost of activity i
τ_i :	per unit tardiness cost of activity i
E_i :	earliness of activity i (integer decision variable)
T_i :	tardiness of activity i (integer decision variable)

of the weighted earliness-tardiness penalty costs of the project activities. In this problem setting, activities have an

In our formulation, 0-1 variables Xijt are defined, which specify whether jth unit of duration of an activity i finishes at time t or not. More specifically, for every unit j of duration of activity i and for every feasible completion time $t \in [EST_i + j, LFT_i - (d_i - j)]$, Xijt is defined as follows:

Xijt = 1, if jth unit of duration of activity i finishes at time t

Xijt = 0, otherwise

Also, 0-1 variables yijt are defined, which specify whether jth unit of duration of an activity i is preempted at time t or not. More specifically, for every unit j of duration of activity i and for every feasible completion time $t \in [EST_i + j, LFT_i - (d_i - j)]$, yijt is defined as follows:

yijt = 1, if jth unit of duration of activity i preempts at time t

yijt = 0, otherwise

The variables Xijt and yijt can only be defined over the time interval of the activity in question. These limits are determined using the traditional forward and backward pass calculations. The backward pass calculation is started from a fixed project deadline T.

Introducing the binary decision variables Xijt and yijt , as well as the integer variables Ei and Ti denoting the earliness and tardiness of activity i ,respectively, and using the above notation, preemptive resource-constrained project scheduling problem with weighted earliness tardiness and preemption penalties (PRCPSPWETPP) under the minimum total early-tardy and preemption penalty cost objective can be mathematically formulated as follows:

$$\min Z = \sum_{i=2}^{n-1}\left(v_i E_i + \tau_i T_i\right) + \sum_{i=2}^{n-1}\left(\pi_i\left(\sum_{j=1}^{d_i}\sum_{t=EFT_i-(d_i-j)}^{LFT_i-(d_i-j)} y_{ijt}\right)\right) \qquad (1)$$

subject to

$$E_i \geq h_i - \sum_{t=EFT_i}^{LFT_i} t\, X_{id_it} \qquad \text{for i = 2,..., n - 1} \qquad (2)$$

$$T_i \geq \sum_{t=EFT_i}^{LFT_i} t\, X_{i d_i t} - h_i \qquad \text{for } i = 2,\ldots, n-1 \tag{3}$$

$$\sum_{t=EFT_i}^{LFT_i} t\, X_{i d_i t} \leq \mathrm{LST}_j \qquad \text{for all } (i, j) \in A \tag{4}$$

$$\sum_{t=EFT_i-(d_i-j+1)}^{LFT_i-(d_i-j+1)} t\, X_{i(j-1)t} + 1 \leq \sum_{t=EFT_i-(d_i-j)}^{LFT_i-(d_i-j)} t\, X_{ijt} \quad \text{for} \begin{cases} i = 2,\ldots, n-1 \\ j = 2,\ldots, d_i \end{cases} \tag{5}$$

$$\sum_{t=EST_i+j}^{LFT_i-(d_i-j)} t\, X_{ijt} = 1 \quad \text{for} \begin{cases} i = 2,\ldots, n-1 \\ j = 1,\ldots, d_i \end{cases} \tag{6}$$

subject to

$$X_{ijt} \leq X_{i(j+1)(t+1)} + y_{ijt} \quad for \begin{cases} i = 2,\ldots, n-1 \\ j = 1,\ldots, d_i \\ t = EFT_i-(d_i-j),\ldots, LFT_i-(d_i-j) \end{cases} \tag{7}$$

$$\sum_{i=2}^{n-1} \sum_{j=1}^{d_i-1} r_i X_{ijt} \leq a_k \quad \text{for} \begin{cases} k = 1 \\ t = 1,\ldots, T \end{cases} \tag{8}$$

$$X_{ijt},\ y_{ijt} \in \{0,1\} \quad \text{for} \begin{cases} i = 2,\ldots, n-1 \\ j = 1,\ldots, d_i \\ k = 1,\ldots, m \\ t = EST_i + j,\ldots, LFT_i-(d_i-j) \end{cases} \tag{9}$$

$$E_i, T_i \in \text{int} \quad \text{for } i = 2,\ldots, n-1 \tag{10}$$

The objective in Eq. (1) is to minimize the total cost of the project. Eq. (2) and (3) compute the earliness and tardiness of each activity. The constraint set given in Eq. (4) imposes the finish-start precedence relations among the activities. In Eq. (5) it is specified that the finish time for every unit of duration of an activity has to be at least one time unit larger than the finish time for the previous unit of duration. Eq. (6) specifies that only one completion time is allowed for every unit of duration of an activity. Eq. (7) guarantee that if two successive units of duration an activity *i* (i.e. unit *j* and *j*+1) are interrupted at time *t*; therefore corresponding decision variable y_{ijt} must set to 1. The resource constraints for every resource type *k* are specified in Eq. (8) by considering for every time instant *t* and every resource type *k*, all possible completion times for every units of duration of all activities *i* such that the activity is in progress in period *t*. This constraint set stipulates that the resource constraints cannot be violated. Eq. (9) and (10) specifies that the decision variables X_{ijt} and y_{ijt} are binary, while E_i and T_i are integer. This formulation requires the definition of at most $2T\sum_{i=1}^{n} d_i$ binary decision variables and of $2(n-2)$ integer variables. Also, the number of constraints of the formulation amounts to at most $2(n-2)+n(n-1)/2+(2+T)\sum_{i=2}^{n-1} d_i + mT$.

4. Numerical example

In this section, we demonstrate the computation of the optimal PRCPSPWETPP solution on a problem instance that is adapted from the Patterson set (Patterson, 1984, 854-867). The corresponding *AON* project network is shown in Fig.1.

There are 7 activities (and two dummy activities) and one resource type with an availability of 5. The number above the node denotes the activity duration, while the numbers below the node denote the due date, the unit early-tardy cost (For ease of representation, we assume the unit earliness costs to equal the unit tardiness costs) and the resource requirements, respectively. Also, we assume the preemption penalty for all activities equal to 1. The optimal non-preemptive schedule for this example is presented in Fig.2.

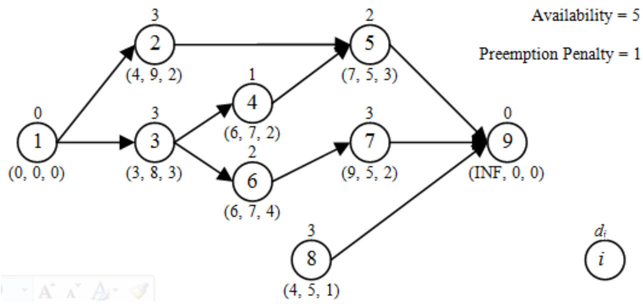

Figure 1. Problem instance for the PRCPSPWETPP

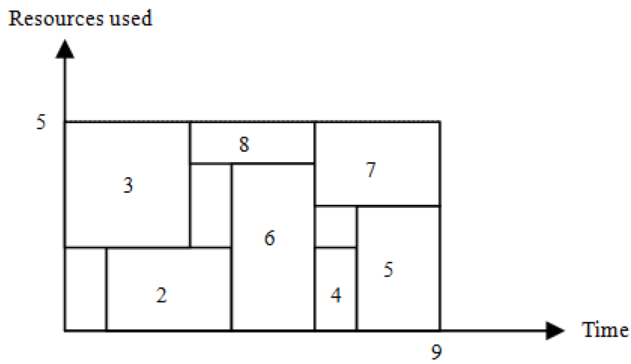

Figure 2. Optimal non-preemptive schedule to the problem example

The proposed formulation for this problem example requires the definition of 129 binary decision variables and of 14 integer variables. The number of constraints and nonzero elements of constraints matrix equals to 108 and 580, respectively. Using the LINGO version 8, based on branch and bound method, we obtained the optimal schedule of Fig.3 with a cost of 23. This problem solved within 1 second of CPU-time. Of course, this schedule is presented at the level of the sub-activities, that is, each activity *i* is divided to d_i segment with duration of 1 and resource requirement of r_i.

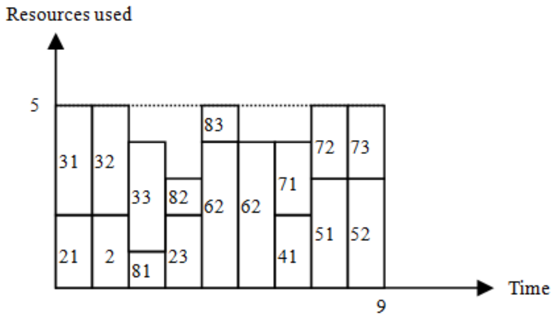

Figure 3. The optimal schedule at the level of the sub-activities

Translating this optimal schedule in terms of the original activities, the optimal preemptive schedule of Fig.4 is obtained. It is should be obvious that in this optimal schedule 2^{th} unit of duration of activity 2 is preempted at time 2.

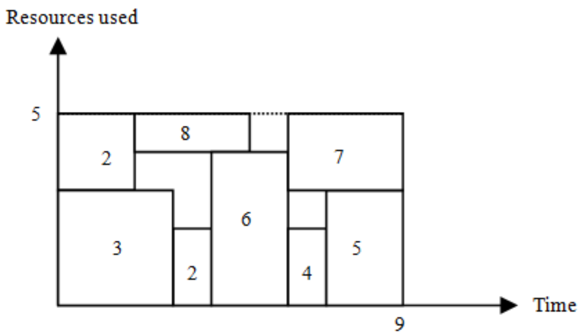

Figure 4. The optimal preemptive schedule at the level of the activities

5. Computational results

In order to validate the integer programming model for the preemptive resource constrained project scheduling problem with preemption penalties and weighted earliness tardiness penalties, a problem set consisting of 900 problem instances was generated. This problem set consisting of equally 300 instances with 10, 20 and 30 activities. The problem set was extended with unit earliness-tardiness penalty costs and preemption penalties for each activity which are randomly generated between 1 and 10. The due dates were generated in the same way as described by Vanhoucke and et al. (Vanhoucke & et al, 2000a, 179-196). First, a maximum due date was obtained for each project by multiplying the critical path length by 1.5. Subsequently, we generate random numbers between 1 and maximum due date. The numbers are sorted and assigned to the activities in increasing order. Activity durations and activity resource demand are randomly selected between 1 and 10. Maximum number of predecessors and successors and number of resource types supposed 3.

Table 1
The average CPU-time and the standard deviation needed to solve the PRCPSPWETPP

Number of activities	Number of problems	Average CPU-time	Standard deviation
10	300	1.29	3.34
20	300	9.54	26.29
30	300	35.09	71.81

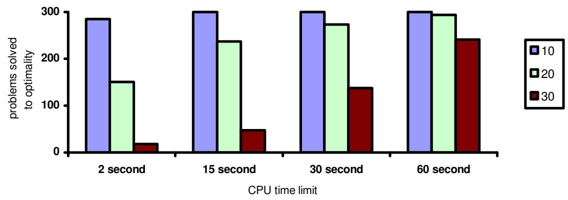

Figure 5. Effect of the number of activities and the allowed CPU-time for the problem

The problem set has solved using the LINGO version 8 under windows XP on a personal computer with Pentium 4, 1.7 GHz processor. Table 1 represents the average CPU-time and its standard deviation in second for a different number of activities with a time limit of 60 second. 95% of problems with 10 activities can be solved to optimality within 2 second of CPU-time. For problems consisting 20 activities, 79% of the problems can be solved to optimality when the allowed CPU-time is 15 second, whereas 91% of the problems can be optimally solved where the CPU-time limit is 30 second. For problems with 30 activities, 46% of the problems can be solved within 30 second of CPU-time whereas 80% of the problems can be solved to optimality when the allowed CPU-time is 60 second. Fig.5 displays the number of problems solved to optimality for a different number of activities and allowed CPU-time.

6. Summary and conclusions

This paper reports on an integer programming based procedure for *p*reemptive *r*esource *c*onstrained *p*roject *s*cheduling *p*roblem with *w*eighted *e*arliness *t*ardiness and *p*reemption *p*enalties (PRCPSPWETPP). The objective is to schedule the activities in order to minimize the total cost of earliness-tardiness and preemption penalties subject to the precedence constraints, resource constraints and a fixed deadline on project. Pure integer programming model applied for solving a numerical example. Finally, some test problems are solved with LINGO version 8 and computational results are reported.

7. References

[1] C. N. Potts, L. N. Van Wassenhove, Integrating scheduling with batching and lot-sizing: A review of algorithms and complexity, Journal of the Operational Research Society 43 (1992) 395-406.

[2] C. L. Monma,C. N. Potts, Analysis of heuristics for preemptive parallel machine scheduling with batch setup times, Journal of Operations Research 41 (1993) 981-993.

[3] B. Chen, A better heuristic for preemptive parallel machine scheduling with batch setup times, SIAM Journal of Computation 22 (1993) 1303-1318.

[4] S. Zdrzalka, Preemptive scheduling with release dates, delivery times and sequence independent setup times, European Journal of Operational Research 76 (1994) 60-71.

[5] P. Schuurman, G. J. Woeginger, Preemptive scheduling with job-dependent setup times, Proceeding of the 10th ACM-SIAM Symposium on Discrete Algorithms (1999) 759-767.

[6] Z. Liu, T. C. E. Cheng, Scheduling with job release dates, delivery times and preemption penalties, Information Processing Letter 82 (2002) 107-111.

[7] F. M. Julien, M. J. Magazine, N. G. Hall, Generalized preemption models for single-machine dynamic scheduling problems, IIE Transactions 29 (1997) 359-372.

[8] M. Vanhoucke, Exact algorithms for various types of project scheduling problems Nonregular objectives and time/cost trade-offs, Unpublished Ph.D. Dissertation, Katholieke Universiteit Leuven (2001).

[9] M. Vanhoucke, E. Demeulemeester, W. Herroelen, An exact procedure for the resource constrained weighted earliness-tardiness project scheduling problem. Annals of Operations Research 102 (2000a) 179-196.

[10] M. Vanhoucke, E. Demeulemeester, W. Herroelen, Maximizing the net present value of a project with progress payments, Research Report 0028 (2000b) Department of Applied Economics, Katholieke Universiteit Leuven.

[11] L. A. Kaplan, Resource constrained project scheduling with preemption of jobs, Unpublished Ph.D Thesis (1988) University of Michigan.

[12] E. Demeulemeester, W. Herroelen, An efficient optimal solution procedure for the preemptive resource constrained project scheduling problem, European Journal of the Operational Research 90 (1996) 334-348.

[13] J. H. Patterson, A comparison of exact procedures for solving the multiple constrained resource project scheduling problem, management science 30 (1984) 854-867.