Journal of Industrial Engineering International *Islamic Azad University, South Tehran Branch*
July 2008, Vol. 4, No. 7, 25-32

A mathematical model for the multi-mode resource investment problem

M. Sabzehparvar*

Assistant Professor, Department of Industrial Engineering, Islamic Azad University, Karaj Branch, Karaj, Iran

S. M. Seyed-Hosseini

Professor, Department of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran

S. Nouri

Assistant Professor, Department of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran

Abstract

This paper presents an exact model for the resource investment problem with generalized precedence relations in which the minimum or maximum time lags between a pair of activities may vary depending on the chosen modes. All resources considered are renewable. The objective is to determine a mode and a start time for each activity so that all constraints are obeyed and the resource investment cost is minimized. Project scheduling of this type occurs in many fields for instance, construction industries. The proposed model has been inspired by the packing problems. In spite of the fact that it needs a feasible solution to start for conventional models, the new model has no need for a feasible solution to startup with. Computational results with a set of 60 test problems have been reported and the efficiency of the proposed model has been analyzed.

Keywords: Multi-mode; Temporal constraints; Time / resource trade-off; Time windows

1. Introduction

In practice we generally can procure or rent as many resource units as we need, but we tend to hire a fixed number of resources of type k, R_k at the start of the project and fired them at the end of the project. Thus the maximum number of resource of any type must be determined.

Let $C_k \geq 0$ denote the procurement cost per unit of resource $k \in R$. Following the classification schemes for project scheduling proposed by [2,3] the time-constrained Resource Investment Problem or ***RIP*** in general case when we have minimum and maximum time lags (temporal constraints) between project activities is denoted by $PS \mid temp, \bar{d} \mid C_k$ or briefly ***RIP-GPR*** in which $\bar{d}$ represents an upper bound on the shortest project duration (deadline) and ***GPR*** represents General Precedence Relations [13, P. 22, 278].

The objective function of this problem tries to minimize total cost of hiring R_k , while in classic well-known Resource-Constrained Project Scheduling Problem or briefly ***RCPSP***, project duration (makespan) is minimized. *RCPSP* in situation considering minimum and maximum time lags denoted by $PS \mid temp \mid C_{\max}$ or ***RCPSP-GPR*** [13, P. 22].

Moreover in *RIP,* the pattern of resource usage over time is much more important than the pick demand of the schedule. In such situations considering alternative ways (modes) of executing of individual activities help to have flexibility to achieve the best pattern of resource usage and minimum level of R_k . These modes differ in processing time, time lags to other activities, and resource requirements. They re-

* *Corresponding author. E-mail:msabzeh@gmail.com*

flect time/resource trade-offs and resource–resource trade-offs [13, P. 160].

When multi-mode is taken into account of the problem *RIP,* it is called Multi-mode Resource Investment Problem or *briefly* ***MRIP*** and in general case denoted by $MPS \mid temp, \bar{d} \mid C_k$ or ***MRIP-GPR***. While the classic type of multi-mode problem is called the Multi-Mode Resource-Constrained Project Scheduling Problem or briefly ***MRCPSP*** and in general case is denoted by $MPS \mid temp \mid C_{\max}$ or ***MRCPSP-GPR*** [13, P. 160].

The focus of this work is to consider a general case in which the associated minimum or maximum time lag may depend on the execution modes of both activities i and j.

2. Problem formulation

Let:

v_{im_i} Binary decision variable; *1* if activity *i* is performed in mode m_i, *0* otherwise.

C_k Procurement cost per unit of resource k $(k = 1,\ldots K)$.

S_i / F_i Starting/finishing time of activity i.

d_{im_i} Duration of activity i in mode m_i.

t Discrete time unit.

$SS_{ij}^{\min} / SS_{ij}^{\max}$ Minimal/maximal time lag between start to start times of activities i and j.

$SF_{ij}^{\min} / SF_{ij}^{\max}$ Minimal/maximal time lag between start to finish times of activities i and j.

$FS_{ij}^{\min} / FS_{ij}^{\max}$ Minimal/maximal time lag between finish to start times of activities i and j.

$FF_{ij}^{\min} / FF_{ij}^{\max}$ Minimal/maximal time lag between finish to finish times of activities i and j.

es_i / ls_i Earliest / latest starting time of activity i.

R_k Maximum number of resource type k available per period.

r_{im_ik} Resource requirement of type k for activity i in mode m_i.

K Number of resource types required for the project.

M_i Number of modes for activity i.

$\bar{d}$ An upper bound on the shortest project duration.

E_{SS}, E_{SF}, E_{FS} and E_{FF} are defined as the resulting set of temporal relations.

Assuming an *AoN* network *N(x)* in standardized form with minimal start to start precedence relations using the transformation rules [2], the problem can be modeled conceptually as follows [13, 16] :

$$\text{Min} \quad \sum_{k \in R} C_k \max r_{im_ik} \quad \text{or} \quad \sum_{K \in R} C_k R_k \tag{1}$$

Subject to:

$$\sum_{m_i \in M_i} v_{im_i} = 1 \qquad (i \in V) \tag{2}$$

$$s_j - s_i \geq \sum_{m_i=1}^{M_i} \sum_{m_j=1}^{M_j} ss_{im_i jm_j}{}^{\min} v_{im_i} . v_{jm_j}$$
$$i < j, \quad \langle i, j \rangle \in E_{ss^{\min}} \tag{3}$$

$$r_k(S,t,v) \leq R_k \qquad (K \in R, \quad 0 \leq t \leq \bar{d}) \tag{4}$$

$$S_0 = 0 \quad , \quad S_i \geq 0 \tag{5}$$
$$x_{im_i} \in \{0,1\} \quad (i \in V, \quad m_i \in M_i)$$

The objective function (1) minimizes the resource investment cost. Eq. (2) ensures that only one mode is selected. Eq. (3) is the *GPR* in standardized form with mode dependent time lags. Eq. (4) expresses that at no time instant of t, during the project horizon between 0 and $\bar{d}$ the resource availability may be violated. Moreover, we define:

$$\bar{d} = \sum_{i \in V} \max(p_i, \max_{\langle i,j \rangle \in \mathrm{E}} \delta_{ij}) \tag{6}$$

which represents an upper bound on the shortest project duration. Let:

$$r_k(s,t,v) := \sum_{i \in A(s,t)} r_{ik} \qquad (k \in R, t \geq 0) \qquad (7)$$

be the amount of resource k used at time t, in which:

$$A(s,t,x), A(s,t,v) := \{i \in V | S_i \leq t < S_i + P_i\} (t \geq 0) \qquad (8)$$

is the set of (real) activities in progress at time t, also called The *active set* at time t.

This non-linear program, however, cannot be solved directly because it is not easy to translate the set $A(s,t,x)$ that is used in Eq. 4 into a mathematical programming formulation. Hence other programming formulations have to be used in order to be able to specify the resource constraints in the correct and solvable form.

The following mathematical programming formulation for *MRIP-GPR,* have been developed by De Reyck and Herroelen [6] based on previous work of Talbot [5: Page 512]. In this formulation all maximal time lags are transformed into equivalent minimal time lags with a negative value in the opposite direction. For instance, a $FS_{ij}^{\max}$ time lag is transformed into a $SF_{ji}^{\min}$ time lag [2]. The decision variables are introduced as follows:

$$x_{im_it} = \begin{cases} 1, & \text{if activity } i \text{ is performed in mode } m_i \text{ and started at time } t \\ 0, & \end{cases}$$

$$\text{Min} \quad \sum_{k \in R} C_k \max r_{im_ik} \quad \text{or} \quad \sum_{K \in R} C_k R_k$$

Subject to:

$$\sum_{m_i=1}^{M_i} \sum_{t=es_i}^{ls_i} x_{im_it} = 1 \qquad i = 1,2,\ldots,n \qquad (9)$$

$$\sum_{m_i=1}^{M_i} \sum_{t=es_i}^{ls_i} (t + SS_{ij}^{\min}) x_{im_it} \leq \sum_{m_j=1}^{M_j} \sum_{t=es_j}^{ls_j} t x_{jm_jt}$$
$$(i,j) \in E_{SS} \qquad (10)$$

$$\sum_{m_i=1}^{M_i} \sum_{t=es_i}^{ls_i} (t + SF_{ij}^{\min}) x_{im_it} \leq \sum_{m_j=1}^{M_j} \sum_{t=es_j}^{ls_j} (t + d_{jm_j}) x_{jm_jt}$$
$$(i,j) \in E_{SF} \qquad (11)$$

$$\sum_{m_i=1}^{M_i} \sum_{t=es_i}^{ls_i} (t + d_{im_i} + FS_{ij}^{\min}) x_{im_it} \leq \sum_{m_j=1}^{M_j} \sum_{t=es_j}^{ls_j} t x_{jm_jt}$$
$$(i,j) \in E_{FS} \qquad (12)$$

$$\sum_{m_i=1}^{M_i} \sum_{t=es_i}^{ls_i} (t + d_{im_i} + FF_{ij}^{\min}) x_{im_it} \leq \sum_{m_j=1}^{M_j} \sum_{t=es_j}^{ls_j} (t + d_{jm_j}) x_{jm_jt}$$
$$(i,j) \in E_{FF} \qquad (13)$$

$$\sum_{i=1}^{n} \sum_{m_i=1}^{M_i} r_{im_ik} \sum_{s=\max\{t-d_{im},es_i\}}^{\min\{t-1,ls_i\}} x_{im_is} \leq R_k$$
$$k = 1,2,\ldots,K; \quad t = 1,2,\ldots,\bar{d} \qquad (14)$$

$$x_{im_it} \in \{0,1\} \; i = 1,\ldots,n \quad m_i = 1,\ldots,M_i, t = es_i,\ldots,ls_i$$

and $R_k \geq 0 \;\; k = 1,\ldots,K$

R_k and x_{im_it} are the decision variables to be determined. Constraints set (9) ensure that each activity is assigned exactly one mode and exactly one starting time. Constraints (10) to (13) denote the *GPRs*. The resource constraints are given in Eqs. (14) and express that at no time instant of t during the project horizon between 0 and $\bar{d}$ the resource availability for each type may be violated.

The variable x_{im_it} can only be defined over the interval between the earliest and latest starting time of the activity in question. These limits are not determined with the use of the traditional forward and backward pass calculation. The calculation of an earliest start schedule, es_i where there are no resource constraints, can be related to the test for existence of a time-feasible schedule. A time-feasible schedule S_T, for *GPR* exists iff *GPR* has no cycle of positive length. A schedule which satisfies the resource constraints is called resource-feasible and denoted by S_R. A schedule which is both time-feasible and resource feasible is called feasible, and $S_T \cap S_R$ is the set of feasible schedules. To establish the model of Talbot et al., we need to have a feasible schedule and $\bar{d}$ must be known. The problem of finding a feasible schedule of the $MPS \mid temp \mid C_{\max}$ *(MRCPSP-GPR)* and $MPS \mid temp, \bar{d} \mid C_k$ *(MRIP-GPR)* are NP-complete [3, 7, 12: Page 165].

3. Relevant literature review

Exact solution procedures for *RIP* without maximum time lags (precedence constraints) have been presented by Möhring (1984) and Demeulemeester (1995) [5,8,13: P. 278].. Drexl and Kimms have devised upper and lower bounds based on lagrange relaxation and column generation techniques [8]. The case of general temporal constraints (time windows) has been discussed by Nübel and Zimmermann and Engelhardt (1998), Nübel (1999), Neumann and Zimmermann (2000) [3,13]. Selle has developed lower bounds based on continuous, surrogate, and lagrange relaxation techniques [13, P. 278].

Since the early eighties, the multi-mode project scheduling problem has been treated by several authors [13, P. 160]. Exact algorithms have been reviewed by Hartmann and Drexl [10]. The most efficient method for solving this problem known thus far is the branch-and-bound algorithm of Sprecher and Drexl [14]. The best heuristic procedure at present, is a genetic algorithm published by Hartmann [10]. For the case of general temporal constraints, three different algorithms have been proposed by De Reyck and Herroelen [6], Dorondorf [7] and Heilmann [11].

In order to be able to specify the resource constraints in the correct and solvable form, the best $0-1$ programming model based on an extension of the formulation by Pritsker et al., has been presented by Talbot [5]. The model of Talbot have been developed by Reyck and Herroelen [6] for the case of general temporal constraints.

4. Mathematical formulation of *RIP-GPR*

The formulations developed have some difficulties in translating the sets of activities which are in progress into linear resource constraints[1,9, 5]. The formulation which is presented here has been inspired by the packing problems models.

There is a certain correspondence between boxes to be packed and activities to be scheduled. In a rather simple approach for *RIP* and *RCPSP* two types of renewable resources is considered. As shown in figure 1, each box would correspond to an activity, with a processing time equal to the length and a resource request of type k (k=1,2) equal to width and height respectively. An empty box B_0 of width W_0 equal to time horizon $\bar{d}$, length L_0 equal to R_1, the resource capacity available of type 1 and height H_0 equal to R_2, the resource capacity available of type 2 is given.

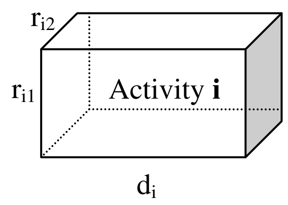


Figure 1. Representation of an activity.

There is a series of boxes B_i (or Activities A_i) ($i=1,...,n$), of width $w_i = d_i$, length $l_i = r_{i1}$ and height $h_i = r_{i2}$ to be packed in which index m_i has been omitted from both d_{im_i} and r_{im_ik} for the case of single mode. Furthermore, the constraint that activity preemption is not allowed corresponds to the natural requirement that boxes must be packed as a whole. The bottom left of the box is placed at $(0,0,0)$ with its six sides parallel to *X*-, *Y*-, and *Z*-axis, respectively.

The x-coordinate of the bottom left behind corner of activity i is given by the activity starting time and is the most important decision variable to be determined. Thus, *GPRs* can be formulated in standard form as follows:

$$x_i + SS_{ij}^{\min} \le x_j \le x_i + SS_{ij}^{\max}$$
$$(i,j) \in E_{SS} \qquad (15)$$

The finishing time of end activities should not be exceeded from $\bar{d}$, i.e.:

$$\bar{d} - x_i - d_i \ge 0 \qquad i \in \textit{end activities} \qquad (16)$$

The constraints for packing boxes are as follows [4]:

Each edge of a box should be parallel to an edge of the main box. There should be no overlapping for any two boxes, i.e., the overlapping area is zero.
The second constraint above for project scheduling must be changed as follows:

There should be no overlapping for any two boxes, toward X- and Y-coordinates as well as X- and Z-coordinates. i.e., it doesn't matter to have overlapping between Y- and Z-coordinates.
The bottom left coordinates of activity A_i are (x_i, y_i, z_i) The top right coordinates of activity A_i are $(x_i + p_i, y_i + r_i^1, z_i + r_i^2)$. We use t_{xji} and t_{xij} to denote whether the activity i to be located at the right hand side of the activity j or vise versa respec-

tively, without any overlapping between them. We use the same notations as t_{yij} and t_{yji} for Y-coordinate, t_{zij} and t_{zji} for Z-coordinate.

Using the binary decision variables above, these constraints can be stated as follows:

$$x_j - x_i - d_i + M * t_{xij} \geq 0$$
$$i = 1,...,n-1,\ j = 2,...,n \quad (17)$$

$$x_i - x_j - d_j + M * t_{xji} \geq 0$$
$$i = 1,...,n-1,\ j = 2,...,n \quad (18)$$

$$y_j - y_i - r_{i1} + M * t_{yij} \geq 0$$
$$i = 1,...,n-1,\ j = 2,...,n \quad (19)$$

$$y_i - y_j - r_{j1} + M * t_{yji} \geq 0$$
$$i = 1,...,n-1,\ j = 2,...,n \quad (20)$$

$$t_{xij} + t_{xji} + t_{yij} + t_{yji} = 3$$
$$i = 1,...,n-1,\ j = 2,...,n \quad (21)$$

where M is a big constant. Equations above ensure that there should be no overlapping for any two boxes, between x- and y-coordinates. For x- and z-coordinates which are applicable in double types of renewable resources, the same equations as (24), (25) and (26) are defined as follows:

$$z_j - z_i - r_{i2} + M * t_{zij} \geq 0$$
$$i = 1,...,n-1,\ j = 2,...,n \quad (22)$$

$$z_i - z_j - r_{j2} + M * t_{zji} \geq 0$$
$$i = 1,...,n-1,\ j = 2,...,n \quad (23)$$

$$t_{xij} + t_{xji} + t_{zij} + t_{zji} = 3$$
$$i = 1,...,n-1,\ j = 2,...,n \quad (24)$$

When a precedence relation of type $FS_{ij}^{\min} \geq 0$ between two activities exists, clearly activity i precedes activity j tx_{ij} is forced to get zero, so that it is not necessary to write Equations (17) to (24).

In this formulation, resource constraints can be formulated as follows:

$$R_1 - y_i - r_{i1} \geq 0 \quad i = 1,...,n \quad (25)$$

$$R_2 - z_i - r_{i2} \geq 0 \quad i = 1,...,n \quad (26)$$

When there are a set of activities without any relation among them, if scheduled in parallel, they would violate resource constraints. In order to resolve a resource conflict between two activities, the location of one of them must be changed as illustrated in Figure 3. We use the minimization of the resource investment cost as Equation (1).

5. Formulation of *MRIP-GPR*

In the case of *MRIP-GPR* individual activities can be executed in alternative ways (modes). Activity i, $(i = 1,...,n)$ when performed in mode m_i, $(m_i = 1,...,M_i)$ has a duration d_{im_i} and requires r_{im_ik}, a constant amount of resource k over duration. To describe the mathematical formulation, Let:

$$d_i = \sum_{m_i=1}^{M_i} d_{im_i} . v_{im_i} \qquad i = 1,...,n \quad (27)$$

$$r_{ik} = \sum_{m_i=1}^{M_i} r_{im_ik} . v_{im_i} \quad i = 1,...,n\ k = 1,...,K \quad (28)$$

MRIP-GPR can be formulated by replacing Equations (27), (28) and (2) into the *RIP-GPR* model.

6. Formulation of *MRIP-GPR with Mode-Dependent time lags*

In this section, a general case of *MRIP-GPR* in which the associated minimal or maximal time lags may depend on the execution modes of both activities i and j is considered. This case is called *MRIP-GPR with Mode-Dependent time lags*. Figure 2 shows the representation of this case in an AoN network in which GPRs are transformed into standard form of minimal start to start precedence relations, using the transformation rules [2]. Each activity can be executed in two alternative modes, say, modes 1 and 2 [13: Page 163].

As shown in Figure 2, each mode of an activity is indicated by an element in the duration vector and a row in the matrix of resource requirements of type k (k=1,2). Clearly number of assignment will be increase exponentially. Assume a network $N(v)$ in standardized form, the weight of an arc $\langle i, j \rangle \in E_{SS}$

(arc set) in multi-mode project network N represents a matrix $SS_{ij}^{\min} = (SS_{im_i jm_j}^{\min})_{m_i \in M_i, m_j \in M_j}$, where the elements $SS_{im_i jm_j}^{\min} \in Z$ denote the (scalar) arc weights that refer to the execution of activity i in mode $m_i \in M_j$ and execution of activity j in mode $m_j \in M_j$. For assignment v,

$$SS_{ij}^{\min}(v) = \sum_{m_i \in M_i} \sum_{m_j \in M_j} SS_{im_i jm_j}^{\min} v_{im_i} . v_{jm_j}$$
$$(i, j) \in E_{SS} \qquad (29)$$

is the resulting weight of arc $\langle i, j \rangle$ in network $N(v)$. An assignment v is called *time-feasible* if $N(v)$ dose not contain any cycle of positive length. A schedule S_T is said to be *time-feasible* with respect to assignment v if S_T satisfies the temporal constraints:

$$S_j - S_i \geq SS_{ij}^{\min}(v) \qquad (i, j) \in E \qquad (30)$$

Equation (30) is nonlinear, in order to keep linearity with these additions and formulate this general problem in linear mixed integer programming, let:

$$q_{im_i, jm_j} = \begin{cases} 1 & \text{if activity i and j are performed in mode } m_i \text{ and } m_j \text{ respectively} \\ 0 & \text{otherwise} \end{cases}$$

Then Equation (15) can be replaced by the following constraints after transforming to the standard form:

$$x_j - x_i - \sum_{m_i \in M_i} \sum_{m_j \in M_j} SS_{im_i, jm_j}^{\min} q_{im_i, jm_j} \geq 0$$

For GPRs transformed to $SS_{im_i, jm_j}^{\min}$ (31)

$$2q_{im_i, jm_j} - v_{im_i} - v_{jm_j} \leq 0$$

For GPRs with a matrix of time lags (32)

in which $SS_{im_i, jm_j}^{\min}$ is the transformed matrix of minimal time lags. The dependency of time lags to the execution modes of both activities i and j is ensured by Equation (32).

7. Computational results

In order to show that the model serves to solve instances of practical size, ProGen/max [14] is used to generate 60 *MRIP-GPR* instances in 12 categories according to the combinations of N $(N = 10,20,30)$, M_i $(M_i = 2,3)$ and K $(K = 2,5)$ using control parameters as given in Table 1. The *order strength OS* is a [0, 1]-normalized measure defined as the number of precedence relations, which is minimum for parallel and maximum for series digraphs [14]. The *resource factor RF* reflects the average portion of resources requested by each activity [15]. Setting *RF* at 1 leads to the most complex resource-constrained project scheduling problem instances. The *resource strength RS* measures the scarcity of the resource availability to the respective requirements [15]. For each category (out of 12), 5 instances have been generated.

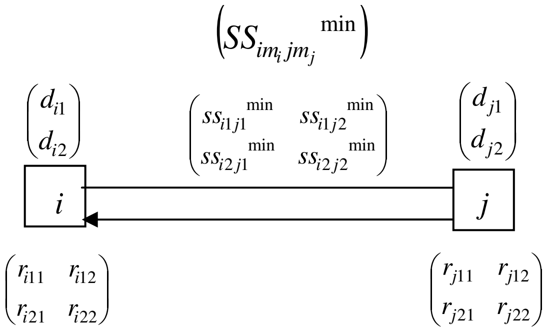
The instances have been optimally solved by the Lingo 8.0 software (http://lindo/lingo8.exe) using branch-and-bound (B&B) method.

Each problem is allowed a maximum of 1000 seconds of CPU time using the Lingo setting (→/Option/General Solver/time Limitation = 1000 Sec.).

All the computational experiments have been carried out on an intel® Celeron® mobile 1.3 GHz Personal Computer with 512 Mb RAM. Since optimal solutions are the same in the two models, Tables 2 and 3 summarize our findings as average CPU times.

8. Conclusion

In this paper we deal with an extension of the resource investment problem where there are multi modes for each activity and general precedence relations with mode dependent time lags. Computational results show that the proposed model is not very sensitive to *number of modes* and in the situations involving $K \leq 3$ is more effective than its competing Talbot et al. model. In addition, it has no need for a feasible solution to startup with.



$$\begin{pmatrix} ss_{i1j1}^{\max} & ss_{i1j2}^{\max} \\ ss_{i2j1}^{\max} & ss_{i2j2}^{\max} \end{pmatrix} \equiv \begin{pmatrix} -ss_{j1i1}^{\min} & -ss_{j1i2}^{\min} \\ -ss_{j2i1}^{\min} & -ss_{j2i2}^{\min} \end{pmatrix}$$

$$\left(SS_{im_i jm_j}{}^{\max}\right) \equiv -\left(SS_{jm_j im_i}{}^{\min}\right)$$

Figure 2. Representation of two activities with the matrixes of modes dependent time lags.

Table 1. The parameter settings of the benchmark problem set.

Symbol	Important Control Parameter	Value
N	Number of non-dummy activities	10, 20, 30
M_i	Number of modes per activity	2, 3
d_{im_i}	Duration of each mode	[1, 10]
	Number of initial and terminal activities	[2, 3]
	Maximum number of predecessors and successors	3
OS	Order strength	0.5
K	Number of renewable resource types	2, 5
r_{im_ik}	Renewable resource demand	[1, 10]
$RF_{\mathrm{Re}\,n}$	Resource factor for renewable resources	1
$RS_{\mathrm{Re}\,n}$	Resource strength for renewable resources	0.5

Table 2. The average CPU time for solving five instances in each category with K=2.

Number of activities	Talbot et al. Model		Proposed Model	
	$M_i = 2$	$M_i = 3$	$M_i = 2$	$M_i = 3$
10	0.125	1.301	0.006	0.009
20	26.801	34.473	7.652	8.891
30	89.924	123.845	19.305	21.650

Table 3. The average CPU time for solving five instances in each category with K=5.

Number of activities	Talbot et al. Model		Proposed Model	
	$M_i = 2$	$M_i = 3$	$M_i = 2$	$M_i = 3$
10	0.170	3.120	2.302	2.780
20	46.370	78.975	137.290	142.530
30	239.21	>1000	>1000	>1000

References

[1] Baptiste, P. and Demassey, S., 2004, Tight LP bounds for resource constrained project scheduling. *OR Spectrum*, 26, 251-262.

[2] Bartuch, M., Möhring, R. H. and Radermacher, F. J., 1988, Scheduling project networks with resource constraints and time windows. *Annals of Operations Research,* 16, 201-240.

[3] Brucker, P. and Knust S., 2000, A linear programming and constraint propagation-based lower bound for RCPSP. *European Journal of Operational Research,* 127, 355-362.

[4] Chen, H. D. and Xu, R., 2007, A new heuristic algorithm for rectangle packing. *Computers & Operation Research*, 34, 3270-3280.

[5] Demeulemeester, E. L. and Herroelen, W. S., 2002, *Project Scheduling, A Research Handbook*, , Kluwer Academic Publishers.

[6] De Reyck, B. and Herroelen W., 1999, The multi-mode resource-constrained project scheduling problem with generalized precedence relations. *European Journal of Operational Research*, 119, 538-556.

[7] Dorndorf, U., 2002, *Project Scheduling with Time Windows, From Theory to Application*. Physica-Verlag Heidelberg New York.

[8] Drexl, A. and Kimms, A., 2001**,** Optimization guided lower and upper bounds for the resource investment Problem. *Journal of Operational Research Society*, 52, 340-351.

[9] Elmaghraby, S. E., 1995, Activity nets: A guided tour through some recent developments. *European Journal of Operational Research,* 82, 383-408.

[10] Hartmann, S. and Drexl, A., 1998, Project Scheduling with multiple modes: A comparison of exact algorithms. *Networks*, 32, 283-298.

[11] Heilmann, R., 2003, A branch-and-bound procedure for the multi-mode resource constrained project scheduling problem with minimum and maximum time lags. *European Journal of Operational Research* 144, 348-365.

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

[13] Neumann K., Schwindt C. and Zimmermann, J., 2003, *Project Scheduling with Time Windows and Scarce Resources*, Springer, Berlin.

[14] Sprecher , A. and Drexl, A., 1998, Multi-mode resource-constrained project scheduling by a simple, general and powerful sequencing algorithm, *European Journal of Operational Research*, 107, 431-450.

[15] Schwindt, C., 1998, *Generation of resource-constrained project scheduling problems subject to temporal constraints*, Report WIOR-543, University of Karlsruhe.

[16] Shadrokh S. and Kianfar F., 2004, *A genetic algorithm for resource investment problem*, Ph.D. thesis, Sharif University of Technology, Iran.