

A mathematical model for the multi-mode resource-constrained project scheduling problem with mode dependent time lags

Majid Sabzehparvar ·
S. Mohammad Seyed-Hosseini

Published online: 24 November 2007
© The Author(s) 2007

Abstract This paper presents an exact model for the multi-mode resource-constrained project scheduling problem with generalized precedence relations in which the minimal or maximal 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 project duration is minimized. Project scheduling of this type occurs in many fields, for instance, construction industries. The proposed model has been inspired by the rectangle 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 start. 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 · Rectangle packing · Resource constrained · Time lags · Time windows

1 Introduction

For many real-life applications of project scheduling, it is possible to perform the individual activities in alternative ways (modes). These modes are different in processing time, time lags to other activities and resource requirements. They reflect time-resource tradeoffs and resource-resource tradeoffs [8]. In addition, the capacities of the renewable resources (e.g., machines and manpower) which are taken up

M. Sabzehparvar (✉) · S.M. Seyed-Hosseini
Department of Industrial Engineering, Iran University of Science and Technology, Narmak,
Tehran, Iran
e-mail: msabzeh@gmail.com

S.M. Seyed-Hosseini
e-mail: seyedhoseini@yahoo.com

by the activities are limited. Such real-life projects can be modeled as instances of the Multi-Mode Resource-Constrained Project Scheduling Problem (MRCPSP). *MRCPSP* in the new situation considered by minimal and maximal time lags or Generalized Precedence Relations called MRCPSP-GPR ([5, p. 512], [6]) denoted by $MPS|temp|C_{\max}$ ([7, p. 103], [12, p. 160]). *MRCPSP* is a generalized version of the standard well-known Resource-Constrained Project Scheduling Problem (RCPSP) which is in GPR denoted by RCPSP-GPR or $PS|temp|C_{\max}$ ([11], [12, p. 22]).

In this paper, all resources are assumed to be renewable with limited amounts because renewable resources are usually distributed uniformly over activity duration (no nonrenewable resources are included), and each activity must be performed in one of several possible modes (with each mode possibly having different activity durations, and different resource requirements). The schedule has to be precedence- and resource-feasible and no activity may be interrupted. A project can be represented by an activity-on-node (AoN) network. The objective is to find an assignment of modes for activities as well as precedence- and resource-feasible starting times of all activities in order that the project duration is minimized.

The problem is strongly NP-hard. Moreover, the problem of finding a feasible solution of the $PS|temp|C_{\max}$ (*RCPSP-GPR*) which is easier than the problem $MPS|temp|C_{\max}$ (*MRCPSP-GPR*) is NP-complete ([3, 7], [12, p. 165]). The complexity of the problem is increased when the time lags between a pair of activities may vary depending on the chosen modes called *MRCPSP-GPR with mode-dependent time lags*.

The remainder of this paper is organized as follows: In Sect. 2, we discuss the relevant literature review. Section 3 gives a well-known formulation based on previous works. A brief reason why this specific formulation is tackled by the authors is discussed in Sect. 4. Geometrical formulations for RCPSP-GPR, MRCPSP-GPR and MRCPSP-GPR with mode-dependent time lags are presented in Sects. 5.1, 5.2 and 5.3, respectively. Computational results are reported in Sect. 6. Section 7 is reserved for our conclusions.

2 Relevant literature review

The multi-mode project scheduling problem with precedence constraints among the activities instead of general temporal constraints has been treated by several authors since the early eighties [12, p. 160]. Exact algorithms have been reviewed and their performances have been tested by Hartmann and Drexl [9]. The most efficient method for solving this problem known so far is the branch-and-bound algorithm of Sprecher and Drexl [15]. The best heuristic procedure at present is a genetic algorithm published by Hartmann [12, p. 160]. For the case of general temporal constraints, three different algorithms have been proposed in literature. The tabu search procedure by De Reyck and Herroelen [6], an extension of the branch-and-bound algorithm by Dorondorf [7] for $PS|temp|C_{\max}$ to a multi-mode case and the enumeration scheme of a branch-and-bound procedure is proposed by Heilmann [10].

Since we are going to present a new formulation for $MPS|temp|C_{\max}$, the focus of the remaining literature review will be on the exact approaches.

In order to be able to specify the resource constraints in the correct and solvable form, linear programming based approaches for *RCPSP* (not *MRCPS*P) have been presented by a number of authors. In formulation by Pritsker et al. (1969), the binary decision variable x_{it} is defined to be 1 if activity i finishes at time instant t , and to be 0 otherwise. This formulation requires the use of at most nT binary decision variables and $O(n^2 + mT)$ restrictions [5, Chap. 6]. Kaplan (1998) and Klein (2000) have developed two different models in which 0–1 variable x_{it} is defined to be 1 if the activity i is in progress in period t , and to be 0 otherwise. These two formulations require the use of at most nT binary decision variables and $O(n^2T)$ restrictions which exceeds the order for the formulation by Pritsker et al. [5, Chap. 6]. For the *MRCPS*P without *GPR*, the best 0–1 programming model based on an extension of the formulation by Pritsker et al. has been presented by Talbot [16]. The model of Talbot has been developed by De Reyck and Herroelen [6] for the case of general temporal constraints. Two totally different formulations for *RCPSP* were presented by Alvarez-Valdes and Tamarit (1993) and Mingozzi et al. (1998) [5, Chap. 6]. The first one is based on the definition of a set *IS* of all minimal resource incompatible sets *S* (minimal forbidden sets). A resource incompatible set is defined to be a set of activities between which no precedence relation exists and they would violate the resource constraints provided that these activities were scheduled in parallel. They have introduced the binary decision variable x_{it} , which equals 1 if activity i precedes activity j and 0 otherwise. This formulation requires the definition of $n(n - 1)$ binary decision variables and of n integer variables. The number of restrictions depends on the number of minimal resource incompatible sets and can be at most $O(2^n)$. Unlike the other formulations, the time horizon in this model can be continuous. The second one (Mingozzi et al.) is based on the notion of *feasible subsets*, i.e., subsets of activities between which no precedence relation is specified and that if scheduled in parallel, do not violate the resource constraints. Based on these feasible subsets, binary decision variables x_{it} and y_{it} can be defined as $x_{it} = 1$, if feasible subset i is processed in period t and 0 otherwise, $y_{it} = 1$, if activity j finishes at time instant t and 0 otherwise. This formulation requires the definition of $O(2^nT)$ binary decision variables and $O(n^2, nT)$ restrictions. Since it is difficult to define minimal resource incompatible and feasible subsets for the last two formulations, respectively, they are less efficient than the first three. Moreover, an extension of them to *MRCPS*P-*GPR* causes the combination of subsets to be increased exponentially. The construction of these two formulations is more relevant in the context of lower bound calculations. Apart from heuristic methods, branch-and-bound is probably the most widely used solution technique for solving both *RCPSP* and *MRCPS*P. Since lower bound calculations play the most important role in branch-and-bound procedures, it has been addressed by recent papers [10]. The best known lower bound for *RCPSP* and *MRCPS*P is currently based on the resolution of several large linear programs [3]. Computational experience by Möhring et al. (2003) has shown that already for medium-sized projects with 100 activities, the resulting linear program cannot be solved in an acceptable amount of computing time [12, p. 75]. Brucker and Knust [3] strengthen one of these relaxations by taking into account time windows for the activities and using column generation to deal with the large number of variables. Moreover, in order to reduce the search space of a problem, the concepts of *the interval capacity consistency tests* and *the*

constraint propagation techniques for the case of time windows have been employed by Dorndorf [7, Chap. 4]. Schwindt (1998) and Baptiste et al. (1999) have shown that intervals $[a, b]$ with minimum residual workload capacity can be determined by investigating $O(|\nu|^2)$ critical intervals [1]. The number of intervals will be increased exponentially with the modes, for the case of $MPS|temp|C_{\max}$; Baptiste and Demasse [1] have introduced several sets of cutting planes and used constraint propagation to tighten the initial formulation of the linear programs and improved some lower bounds for *PSPLIB* instances (available on <http://PSPLIB.com>).

3 Problem formulation

Let:

- 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_k} 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{T} an upper bound on the project duration
- T the shortest project duration.

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

In the absence of both resource constraints and multi-mode, a simple linear model can be written as follows [6]:

$$\text{Minimize } F_n \quad (1)$$

$$S_i + SS_{ij}^{\min} \leq S_j \leq S_i + SS_{ij}^{\max}, \quad \langle i, j \rangle \in E_{SS}, \quad (2)$$

$$S_i + SF_{ij}^{\min} \leq F_j \leq S_i + SF_{ij}^{\max}, \quad \langle i, j \rangle \in E_{SF}, \quad (3)$$

$$F_i + FS_{ij}^{\min} \leq S_j \leq F_i + FS_{ij}^{\max}, \quad \langle i, j \rangle \in E_{FS}, \quad (4)$$

$$F_i + FF_{ij}^{\min} \leq F_j \leq F_i + FF_{ij}^{\max}, \quad \langle i, j \rangle \in E_{FF}, \quad (5)$$

$$S_1 = 0, \quad S_i \geq 0, \quad F_i \geq 0, \quad i = 1, 2, \dots, n. \quad (6)$$

In which E_{SF} , E_{FS} and E_{FF} can be transformed into equivalent standard form, E_{SS} using (7).

$$F_i = S_i + d_i, \quad i = 1, 2, \dots, n. \quad (7)$$

In the situations involving resource constraints, a 0–1 programming formulation must be used. This is because there is no easy way in translating the sets of activities which are in progress into linear resource constraints [5, p. 208].

The following mathematical programming formulation for *MRCPSp-GPR*, *MPS|temp|C_{max}* has been developed by De Reyck and Herroelen [6] based on the previous work of Talbot [16]. 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:

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

$$\text{Min} \quad \sum_{t=es_n}^{ls_n} t x_{n1t} \quad (8)$$

subject to:

$$\sum_{m_i=1}^{M_i} \sum_{t=es_i}^{ls_i} x_{im_i t} = 1, \quad i = 1, 2, \dots, n, \quad (9)$$

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

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

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

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

$$\sum_{i=1}^n \sum_{m_i=1}^{M_i} r_{imk} \sum_{s=\max\{t-d_{im}, es_i\}}^{\min\{t-1, ls_i\}} x_{im_i s} \leq R_k, \quad k = 1, 2, \dots, K; \quad t = 1, 2, \dots, T, \quad (14)$$

$$x_{imt} \in \{0, 1\} \quad i = 1, 2, \dots, n, \quad m_i = 1, 2, \dots, M_i, \quad t = es_i, \dots, ls_i. \quad (15)$$

The objective function (8) minimizes the project duration (makespan) in which x_{n1t} is the starting time of the nonreal end activity with a single mode and zero duration. 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 (14) and express that at no time instant of t during the project horizon between 0 and T the resource availability for each type may be violated.

The variable x_{imt} 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 and De Reyck et al., we need to have a feasible schedule and R_k must be known. The problem of finding a feasible schedule of the $PS|temp|C_{\max}$ (*RCPSP-GPR*) and $MPS|temp|C_{\max}$ (*MRCPSP-GPR*) is NP-complete ([3, 7], [12, p. 165]).

4 Difficulties of modeling resource constraints

Resource constraints introduced by (14) are the most difficult constraints. Typical formulations have some difficulties in translating the sets of activities which are in progress into linear resource constraints. Furthermore, most of them need to have a feasible schedule. The difficulty increases when dealing with different and smaller duration units (e.g., hour, minute and second) is necessary. Consequently, the number of decision variables will be increased exponentially. Hence, several programming formulations have to be used in order to be able to specify the resource constraints in the correct and solvable form [5, p. 208].

5 Geometrical formulation

The formulation which is presented here has been inspired by the rectangle packing problems models. First, we outline formal similarities and differences between rectangle packing problems and *RCPSP* [14]. Secondly, we develop a geometrical model for *MRCPSP-GRP* and *MRCPSP-GRP with mode-dependent time lags*.

5.1 Geometrical formulation or *RCPSP*

5.1.1 Main idea

For visualizing the problem in three dimensions, imagine the number of resource types are two ($K = 2$). It can, however, be beyond the confines of two, i.e., any

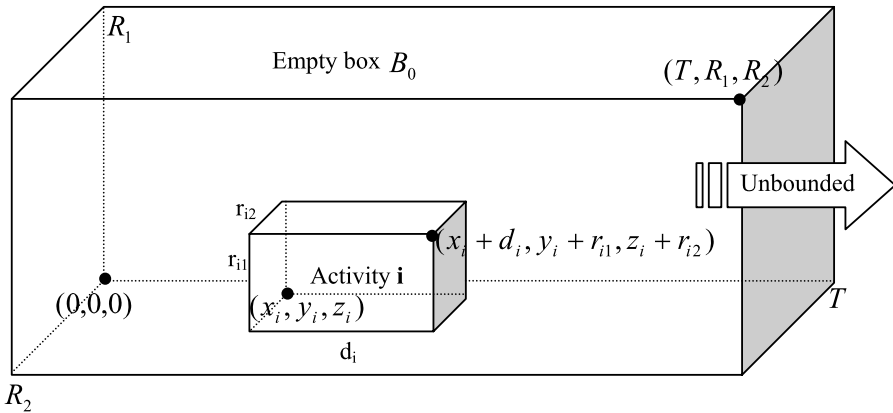


Fig. 1 Geometrical representation of a single mode activity to be packed in an empty box

integer value ($k = 1, 2, 3, \dots$). Moreover, it must be noted that the geometrical model is applicable in the situations involving the uniformly distributed resource needs over processing times. In such a case, there is a certain correspondence between boxes to be packed and activities to be scheduled. As can be seen from Fig. 1, each box would correspond to an activity with a duration 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 T , 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. There is a series of boxes B_i (or Activities A_i) ($i = 1, \dots, n$), 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 $r_{im_i k}$ for the case of a single mode. Furthermore, the constraint that activity preemption is not allowed corresponds to the natural requirement that boxes must be packed as a whole. As shown in Fig. 1, the bottom left behind corner of the big empty box B_0 is supposed to be at point $(0, 0, 0)$ so the top right front corner is (T, R_1, R_2) . Let:

(x_i, y_i, z_i) : the bottom left behind coordinates of activity i ,

$(x_i + d_i, y_i + r_{i1}, z_i + r_{i2})$: the top right front coordinates of activity i .

5.1.2 GPRs constraints

The x -coordinate of the bottom left behind corner of activity i is given by the activity starting time:

$$x_i = S_i, \quad i = 1, 2, \dots, n \quad (16)$$

and is the most important decision variable to be determined. Thus, GPRs can be formulated easily by replacing (16) into (2) to (7) as follows:

$$x_i + SS_{ij}^{\min} \leq x_j \leq x_i + SS_{ij}^{\max}, \quad \langle i, j \rangle \in E_{SS}, \quad (17)$$

$$x_i + SF_{ij}^{\min} \leq x_j + d_j \leq x_i + SF_{ij}^{\max}, \quad \langle i, j \rangle \in E_{SF}, \quad (18)$$

$$x_i + d_i + FS_{ij}^{\min} \leq x_j \leq x_i + d_i + FS_{ij}^{\max}, \quad \langle i, j \rangle \in E_{FS}, \quad (19)$$

$$x_i + d_i + FF_{ij}^{\min} \leq x_j + d_j \leq x_i + d_i + FF_{ij}^{\max}, \quad \langle i, j \rangle \in E_{FF}. \quad (20)$$

5.1.3 Constraint of makespan

Suppose an empty box of variable width (time horizon) T , which is a real decision variable is given. Thus, the finishing time of end activities $x_i + d_i$ should not be exceeded from T , i.e.:

$$T - x_i - d_i \geq 0, \quad i \in \text{end activities}. \quad (21)$$

5.1.4 The objective function

The objective function of the rectangle packing problems which try to maximize the area usage is a nonlinear function. By contrast, we use the minimization of makespan for RCPSP which is linear and the most popular objective in the project scheduling problems as follows:

$$\text{Minimize } T. \quad (22)$$

5.1.5 No overlapping constraints

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

1. Since activity boxes may not be rotated, each edge of an activity box should be parallel to a specific edge of the main box B_0 .
2. There should be no overlapping for any two small boxes, i.e., the overlapping area is zero.

5.1.6 Differences between constraints of project scheduling and packing problem

In the situations involving single type renewable resources in project ($K = 1$), the problem can be formulated as the same as rectangle packing problem. The difficulties start when the number of resource types is two or more ($K \geq 2$). In this case which is the contribution of this paper, the problem must be formulated totally different from the packing problem. In the packing problem, boxes must be packed to a container in which no overlapping between a pair of boxes coordinates is permitted. The no overlapping constraints for project scheduling must be changed as: There should be no overlapping for any two boxes between x- and y-coordinates as well as x- and z-coordinates, i.e., as shown in Fig. 2. It is not important to have overlapping between y- and z-coordinates. This is because the resulting resource profile in the form of the Gantt chart must be determined from the x-y perspective (toward z facing) for $k = 1$, and x-z perspective (toward y facing) for $k = 2$ (see Figs. 4, 5).

For RCPSP with a single type renewable resource in which no precedence relation of type $FS_{ij}^{\min} \geq 0$ between two activities i and j exists, one of the following constraints must be held:

$$(x_j \geq x_i + d_i) \vee (x_i \geq x_j + d_j) \vee (y_j \geq y_i + r_{i1}) \vee (y_i \geq y_j + r_{j1}). \quad (23)$$

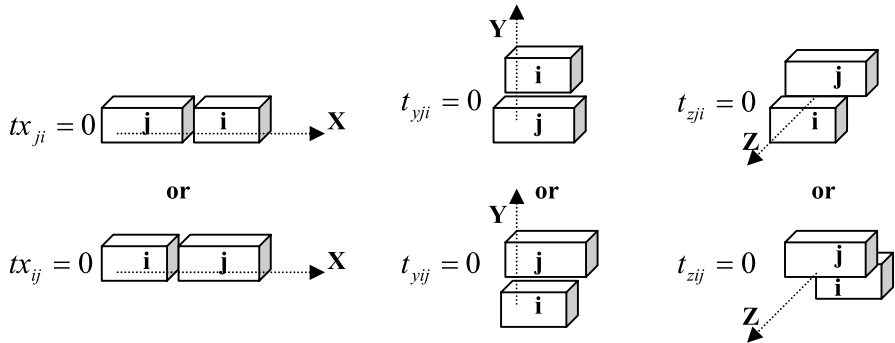


Fig. 2 Permitted no overlapping alternatives to provide Gantt chart and resolve resource conflicts

Let:

t_{xij} : 0–1 integer variable; 0 if activity i to be located at the left-hand side of activity j (activity i precedes activity j) without any overlapping between them, 1 otherwise.

t_{xji} : 0–1 integer variable; 0 if activity j to be located at the left-hand side of activity i (activity j precedes activity i) without any overlapping between them, 1 otherwise.

We use the same definitions for notations t_{yij} and t_{yji} toward y-coordinate, t_{zij} and t_{zji} toward z-coordinate as shown in Fig. 2.

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

$$x_j - x_i - d_i + M * t_{xij} \geq 0, \quad i < j, i = 1, \dots, n-1, j = 2, \dots, n, \quad (24)$$

$$x_i - x_j - d_j + M * t_{xji} \geq 0, \quad i < j, i = 1, \dots, n-1, j = 2, \dots, n, \quad (25)$$

$$y_j - y_i - r_{i1} + M * t_{yij} \geq 0, \quad i < j, i = 1, \dots, n-1, j = 2, \dots, n, \quad (26)$$

$$y_i - y_j - r_{j1} + M * t_{yji} \geq 0, \quad i < j, i = 1, \dots, n-1, j = 2, \dots, n, \quad (27)$$

$$t_{xij} + t_{xji} + t_{yij} + t_{yji} = 3, \quad i < j, i = 1, \dots, n-1, j = 2, \dots, n \quad (28)$$

where M is a big constant. The above equations 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 (26), (27) and (28) are defined as follows:

$$z_j - z_i - r_{i2} + M * t_{zij} \geq 0, \quad i < j, i = 1, \dots, n-1, j = 2, \dots, n, \quad (29)$$

$$z_i - z_j - r_{j2} + M * t_{zji} \geq 0, \quad i < j, i = 1, \dots, n-1, j = 2, \dots, n, \quad (30)$$

$$t_{xij} + t_{xji} + t_{zij} + t_{zji} = 3, \quad i < j, i = 1, \dots, n-1, j = 2, \dots, n. \quad (31)$$

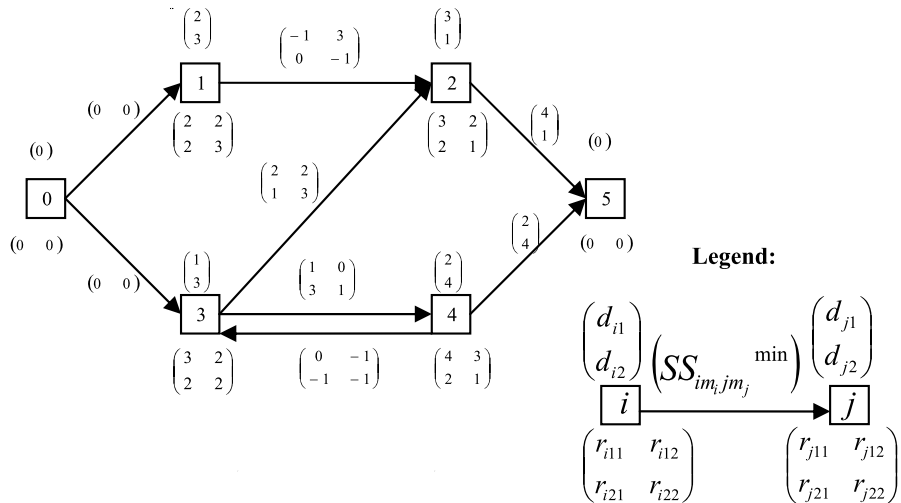


Fig. 3 An example in the form of AoN network

The same equations as (29), (30) and (31) can be developed in situations involving $K > 2$.

When a precedence relation of type $FS_{ij}^{\min} \geq 0$ between two activities exist, clearly activity i precedes activity j by the first part of (19) and tx_{ij} is forced to get zero, so that it is not necessary to write (24) to (31). Since no overlapping constraints set must be written for each pair of activities without relation of $FS^{\min} \geq 0$ and $i < j$, $i = 1, \dots, n-1$, $j = 2, \dots, n$, the number of these constraints sets for all activities in project is $C_{n-NFS^{\min}}^2$, in which NFS^{\min} is the number of non-negative minimum time lags of type finish to start.

5.1.7 Resource constraints

In this geometrical formulation, resource constraints can be formulated easily by supposing an empty box B_0 of length L_0 equal to R_1 , and height H_0 equal to R_2 . So the y- and z-coordinates of activities should not exceed from R_1 and R_2 , respectively. This can be stated in the form of constraints as follows:

$$R_1 - y_i - r_{i1} \geq 0, \quad i = 1, \dots, n, \quad (32)$$

$$R_2 - z_i - r_{i2} \geq 0, \quad i = 1, \dots, n. \quad (33)$$

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 Fig. 3.

5.2 Geometrical formulation for MRCPSP-GPR

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

$$d_i = \sum_{m_i=1}^{M_i} d_{im_i} \cdot v_{im_i}, \quad i = 1, \dots, n, \quad (34)$$

$$r_{ik} = \sum_{m_i=1}^{M_i} r_{im_i k} \cdot v_{im_i}, \quad i = 1, \dots, n, \quad k = 1, \dots, K, \quad (35)$$

$$\sum_{m_i=1}^{M_i} v_{im_i} = 1, \quad i = 1, \dots, n \quad (36)$$

in which v_{im_i} are the binary decision variables with the following definition:

$$v_{im_i} = \begin{cases} 1, & \text{if activity } i \text{ is performed in mode } m_i \\ 0, & \text{otherwise.} \end{cases}$$

MRCPSP-GPR can be formulated by adding (34), (35) and (36) into the *RCPS*P model.

5.3 Geometrical formulation for MRCPSP-GPR with mode-dependent time lags

5.3.1 Problem description

In this section, a general case of *MRCPSP-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 *MRCPSP-GPR with mode-dependent time lags*.

Figure 3 shows the project network N for a project with four real activities 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, e.g., modes 1 and 2 [12, p. 163].

As shown in the legend of Fig 3, 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, for this small problem, there are $2^4 = 16$ different mode assignments.

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 $SS_{im_i jm_j}^{\min} \in \mathbb{Z}$ 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 ,

$$SS_{ij}^{\min}(v) = \sum_{m_i \in M_i} \sum_{m_j \in M_j} SS_{im_i jm_j}^{\min} v_{im_i} \cdot v_{jm_j}, \quad ((i, j) \in E_{SS}) \quad (37)$$

is the resulting weight of arc $\langle i, j \rangle$ in network $N(v)$. An assignment v is called *time-feasible* if $N(v)$ does 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:

$$S_j - S_i \geq SS_{ij}^{\min}(v), \quad (\langle i, j \rangle \in E). \quad (38)$$

Equation (37) is nonlinear in order to keep linearity with these additions and formulate this general problem in linear mixed integer programming. Let:

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

then (17) to (20) 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} , (39)

$$2q_{im_i, jm_j} - v_{im_i} - v_{jm_j} \leq 0 \quad \text{for GPRs with a matrix of time lags} \quad (40)$$

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 (40).

5.3.2 The final model of the MRCPSP-GPR with mode-dependent time lags

Assume all maximal time-lags are transformed into equivalent minimal time-lags in the opposite direction, the mixed integer linear programming formulation for $K = 2$ can be written as follows:

$$\text{Min } T \quad (22)$$

$$T - x_i - \sum_{m_i}^{M_i} d_{im_i} \cdot v_{im_i} \geq 0, \quad i = 1, \dots, n, \quad (41)$$

$$R_1 - y_i - \sum_{m_i}^{M_i} r_{im_i 1} \cdot v_{im_i} \geq 0, \quad i = 1, \dots, n, \quad (42)$$

$$R_2 - z_i - \sum_{m_i}^{M_i} r_{im_i 2} \cdot v_{im_i} \geq 0, \quad i = 1, \dots, n, \quad (43)$$

$$x_j - x_i - \sum_{m_i=1}^{M_i} d_{im_i} \cdot v_{im_i} + M * t_{xij} \geq 0, \quad i < j, \quad i = 1, \dots, n-1, \quad j = 2, \dots, n, \quad (44)$$

$$x_i - x_j - \sum_{m_j=1}^{M_j} d_{jm_j} \cdot v_{jm_j} + M * t_{xji} \geq 0, \quad i < j, \quad i = 1, \dots, n-1, \quad j = 2, \dots, n, \quad (45)$$

$$y_j - y_i - \sum_{m_i=1}^{M_i} r_{im_i1} \cdot v_{im_i} + M * t_{yij} \geq 0, \quad i < j, \quad i = 1, \dots, n-1, \quad j = 2, \dots, n, \quad (46)$$

$$y_i - y_j - \sum_{m_j=1}^{M_j} r_{jm_j1} \cdot v_{jm_j} + M * t_{yji} \geq 0, \quad i < j, \quad i = 1, \dots, n-1, \quad j = 2, \dots, n, \quad (47)$$

$$z_j - z_i - \sum_{m_i=1}^{M_i} r_{im_i2} \cdot v_{im_i} + M * t_{zij} \geq 0, \quad i < j, \quad i = 1, \dots, n-1, \quad j = 2, \dots, n,$$

$$z_i - z_j - \sum_{m_j=1}^{M_j} r_{jm_j2} \cdot v_{jm_j} + M * t_{zji} \geq 0, \quad i < j, \quad i = 1, \dots, n-1, \quad j = 2, \dots, n, \quad (48)$$

(28), (31), (36), (39), (40)

$$t_{xij}, t_{xji}, t_{yij}, t_{yji}, t_{zij}, t_{zji} = \{0, 1\}, \quad i < j, \quad i = 1, \dots, n-1, \quad j = 2, \dots, n, \quad (49)$$

$$v_{im_i}, q_{im_i, jm_j} = \{0, 1\} \quad \text{for GPRs with a matrix of time lags,} \quad (50)$$

$$x_i, y_i, z_i \geq 0, \quad i = 1, \dots, n. \quad (51)$$

In the situations involving $K > 2$, the problem can be formulated simply by replacing $x1_i$ instead of x_i , $x2_i$ instead of y_i , $x3_i$ instead of z_i , \dots , $x(K+1)_i$ instead of dimension $(K+1)$ and so on.

5.3.3 A small example

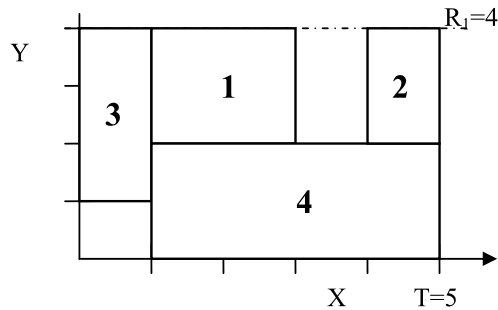
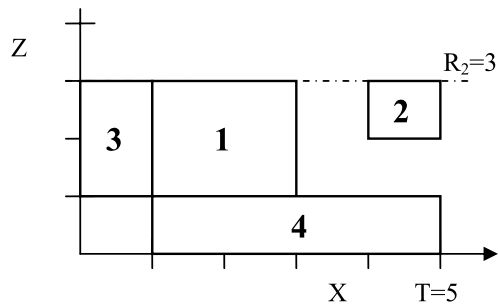
We return to the project and relevant data of Fig. 1. Suppose $R_1 = 4$, $R_2 = 3$. The problem has been solved using *LINGO* 8. Table 1 shows the amount of important decision variables.

Gantt charts in the form of resource profiles of type 1 and 2 are displayed in Figs. 4 and 5, respectively, in which the data in Table 1 is used.

As shown in Figs. 4 and 5, the x-coordinates of activities in both figures are the same.

Table 1 Optimum solution of example 1

Activity	Mode	X	Y	Z
1	1	1	2	1
2	2	4	2	3
3	1	0	1	0
4	2	1	0	0

Fig. 4 Gantt chart in the form of resource profile of type 1**Fig. 5** Gantt chart in the form of resource profile of type 2

6 Computational results

In order to show that the model serves to solve instances of practical size, ProGen/max [14] is used to generate 60 *MRCPSG-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 the control parameters as given in Table 2. 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 [13]. The *resource factor* RF reflects the average portion of resources requested by each activity [13]. 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 [13]. For each category (out of 12), 5 instances have been generated. The instances have been optimally solved by the Lingo 8.0 software (<http://lingo/lingo8.exe>) using branch-and-bound (B&B) method. Since for the model

Table 2 The parameter settings of the benchmark problem set

Symbol	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
$r_{im_i k}$	Renewable resource demand	[1, 10]
RF_{Ren}	Resource factor for renewable resources	1
RS_{Ren}	Resource strength for renewable resources	0.5

Table 3 The average CPU time for solving five instances in each category with $K = 2$

Number of activities	Talbot and De Reyck model		Geometrical 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 4 The average CPU time for solving five instances in each category with $K = 5$

Number of activities	Talbot and De Reyck model		Geometrical 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

of Talbot and De Reyck a feasible solution is required, the following procedure to find a feasible solution is given:

Step 1: Set \bar{T} to an arbitrary number.

Step 2: Calculate earliest and latest starting times using Floyd–Warshal algorithm.

Step 3: Solve the mathematical program using specified \bar{T} , es_i and ls_i . Stop: if the problem is feasible, go to Step 1: otherwise.

Each problem is allowed a maximum of 1,000 seconds of CPU time using the Lingo setting (\rightarrow /Option/General Solver/time Limitation = 1,000 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 3 and 4 summarize our findings as average CPU times.

Table 5 The average CPU time for solving five instances with $K = 2$ and $M_i = 2$

Number of activities	Talbot and De Reyck model		Geometrical model	
	Time unit	Time unit	Time unit	Time unit
	(days)	(hours)	(days)	(hours)
10	0.125	169.508	0.006	0.006

The proposed geometrical model is not very sensitive to M_i and in the situations involving $K \leq 3$ is more effective than its competing Talbot and De Reyck et al. model. In addition, it has no need to have a feasible solution to startup with, so that it can be formulated easily. Also, it must be noted that the geometrical model is applicable in situations in which the resource needs over processing times are uniformly distributed.

In order to analyze the impact of time unit, t on the average CPU time, a simple category ($N = 10$, $M_i = 2$, $K = 2$) has been selected. Assuming duration and planning horizon in hours and having 8 working hours per day, the five previous instances have been optimally solved again. As indicated in Table 5, dealing with smaller duration units (e.g., hour, minute and second) in the Talbot and De Reyck model causes to increase the number of decision variables, constraints and CPU time exponentially, but this is not the case in the geometrical model.

7 Conclusion

In this paper, we considered the multi-mode resource constrained project scheduling problem in which the temporal time lags between a pair of activities may vary depending on the chosen modes. This problem called *MRCPSP-GPR* with mode-dependent time lags is a general case of the problem *MRCPSP-GPR*. Although considering that multi-mode and mode-dependent time lags increase the complexity of the problem, it allows flexibility to achieve an active pattern of resource usage over time as well as a shorter makespan.

First, we presented an exact model for the *RCPSG-GPR* and *MRCPSG-GPR*. Secondly, we developed the model for optimally solving the problem *MRCPSPMCM-GPR* with mode-dependent time lags.

Unlike the typical models, the proposed model has no need for a feasible solution to startup with. Furthermore, time horizon can be continuous in this model thus dealing with different processing time units (e.g., hours, days, weeks, and so on) is possible and has no effect on CPU time.

Future efforts will be devoted to formulate the general case of multi-mode time-constrained resource investment problem or briefly *MRIP-GPR*.

Acknowledgements The authors are grateful to the respected referees for their valuable comments.

Open Access This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

References

1. Baptiste P, Demassey S (2004) Tight LP bounds for resource constrained project scheduling. *OR Spectr* 26:251–262
2. Bartuch M, Möhring RH, Radermacher FJ (1988) Scheduling project networks with resource constraints and time windows. *Ann Oper Res* 16:201–240
3. Brucker P, Knust S (2000) A linear programming and constraint propagation-based lower bound for RCPSP. *Eur J Oper Res* 127:355–362
4. Chen HD, Xu R (2007) A new heuristic algorithm for rectangle packing. *Comput Oper Res* 34:3270–3280
5. Demeulemeester EL, Herroelen WS (2002) Project scheduling, a research handbook, ISBN 1-4020-7051-9. Kluwer
6. De Reyck B, Herroelen W (1999) The multi-mode resource-constrained project scheduling problem with generalized precedence relations. *Eur J Oper Res* 119:538–556
7. Dorndorf U (2002) Project scheduling with time windows, from theory to application, ISBN 3-7908-1516-0. Physica, Heidelberg
8. Elmaghraby SE (1995) Activity nets: A guided tour through some recent developments. *Eur J Oper Res* 82:383–408
9. Hartmann S, Drexl A (1998) Project scheduling with multiple modes: A comparison of exact algorithms. *Networks* 32:283–298
10. Heilmann R (2003) A branch-and-bound procedure for the multi-mode resource constrained project scheduling problem with minimum and maximum time lags. *Eur J Oper Res* 144:348–365
11. Kolisch R, Sprecher A, Drexl A (1995) Characterization and generation of a general class of resource-constrained project scheduling problems. *Manag Sci* 41:1693–1703
12. Neumann K, Schwindt C, Zimmermann J (2003) Project scheduling with time windows and scarce resources, ISBN 3-540-40125-3. Springer, Berlin
13. Schwindt C (1998) Generation of resource-constrained project scheduling problems subject to temporal constraints. Report WIOR-543, University of Karlsruhe
14. Seyed-Hosseini SM, Sabzehparvar M (2007) A geometrical model for the multi-mode resource constrained project scheduling problem. In: Business research conference for international academy of business disciplines, Florida, USA
15. Sprecher A, Drexl A (1998) Multi-mode resource-constrained project scheduling by a simple, general and powerful sequencing algorithm. *Eur J Oper Res* 107:431–450
16. Talbot FB (1982) Resource-constrained project scheduling problem with time-resource trade-offs: the nonpreemptive case. *Manag Sci* 28:1197–1210