Computers & Operations Research 40 (2013) 1060–1063

Contents lists available at SciVerse ScienceDirect

Computers & Operations Research

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

# A note on "event-based MILP models for resource-constrained project scheduling problems"

Christian Artigues [a,b,*], Peter Brucker [c], Sigrid Knust [c], Oumar Koné [d], Pierre Lopez [a,b], Marcel Mongeau [e]

[a] CNRS, LAAS, 7 avenue du Colonel Roche, F-31400 Toulouse, France
[b] Univ de Toulouse, LAAS, F-31400 Toulouse, France
[c] Department of Mathematics/Computer Science, University of Osnabrück, Germany
[d] Laboratoire de Mathématiques et Informatique, UFR-SFA, Université d'Abobo - Adjamé, BP 801 Abidjan 02, Côte d'Ivoire, France
[e] École Nationale de l'Aviation Civile, 7 av. É.-Belin - BP 54005, 31055 Toulouse Cedex 4, France

ARTICLE INFO

Available online 1 November 2012

*Keywords:*
Resource-constrained project scheduling
MILP formulation
Event

ABSTRACT

Recently, new mixed integer linear programming formulations for the resource-constrained project scheduling problem were proposed by Koné et al. [3]. Unfortunately, the presentation of the first new model (called start/end-based formulation SEE) was not correct. More precisely, a set of necessary constraints representing the relative positioning of start and end events of activities was unintentionally omitted in the paper although it was present in the integer program used for the computational experiments. After presenting a counterexample showing the incorrectness, we provide a disaggregated and an aggregated variant of the set of necessary constraints, the disaggregated formulation yielding in theory a better linear programming relaxation. We present computational results showing that although the linear programming relaxations of both formulations yield equivalently poor lower bounds, the disaggregated formulation shows in average a better performance for integer solving of a well-known set of 30-activity instances.



## 1. Introduction

The resource-constrained project scheduling problem (RCPSP) may be formulated as follows. Given are $n$ activities $A=\{1,\ldots,n\}$ and $m$ renewable resources $R=\{1,\ldots,m\}$. A constant number of $B_k$ units of resource $k$ is available at any time. Activity $i\in A$ must be processed for $p_i$ time units and occupies $b_{ik}$ units of resource $k$ during this time period. Furthermore, a set $E$ of precedence relations $(i,j)$ is given, where $(i,j)\in E$ means that activity $j$ cannot start before activity $i$ is completed.

The objective is to determine starting times $S_i$ for the activities $i\in A$ such that

- at each time the total resource demand is less than or equal to the resource availability of each resource $k\in R$,
- the given precedence constraints are fulfilled, i.e. $S_i+p_i\leq S_j$ for $(i,j)\in E$, and
- the makespan $C_{\max}=\max_{i=1}^{n}\{C_i\}$ is minimized, where $C_i:=S_i+p_i$ denotes the completion time of activity $i$.

Additionally, two dummy activities 0 and $n+1$ are introduced indicating the start and the end of the project, respectively. These dummy activities need no resources and have processing time zero. Furthermore, we have $(0,i)\in E$ for all activities $i$ without any predecessor and $(i,n+1)\in E$ for all activities $i$ without any successor. Then $S_{n+1}$ may be interpreted as the makespan of the project.

Recently, two new mixed integer linear programming (MILP) formulations for the RCPSP based on events were introduced in Koné et al. [3]. In the following, we will show that the start/end-based formulation SEE is not correct by presenting a counterexample. Afterwards we give a corrected version and present some computational results.

## 2. The SEE-formulation

The start/end event-based formulation SEE uses the notion of events which correspond to times where an activity starts or ends. The SEE-formulation relies on the fact that for the RCPSP always an optimal left-shifted (semi-active) schedule exists in which the start time of any activity is either 0 or coincides with the completion time of another activity. Therefore, for $n$ activities at most $n+1$ events have to be considered. Let $\mathcal{E}=\{0,1,\ldots,n\}$ be

* Corresponding author at: CNRS, LAAS, 7 avenue du Colonel Roche, F-31400 Toulouse, France.
*E-mail addresses:* christian.artigues@laas.fr (C. Artigues), peter.brucker@uni-osnabrueck.de (P. Brucker), sigrid.knust@uni-osnabrueck.de (S. Knust), mr.okone@gmail.com (O. Koné), pierre.lopez@laas.fr (P. Lopez), marcel.mongeau@enac.fr (M. Mongeau).

0305-0548/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cor.2012.10.018

the index set of the events corresponding to the starting and completion times of the activities.

In the SEE-formulation, two types of binary variables and two types of continuous variables are used. There are binary variables $x_{ie}$, $y_{ie} \in \{0,1\}$ for $i \in A$, $e \in \mathcal{E}$ with

$$x_{ie} := \begin{cases} 1, & \text{if activity } i \text{ starts at event } e \\ 0, & \text{otherwise} \end{cases}$$

and

$$y_{ie} := \begin{cases} 1, & \text{if activity } i \text{ ends at event } e \\ 0, & \text{otherwise} \end{cases}$$

The continuous variables $t_e$ for $e \in \mathcal{E}$ represent the dates of the events $e$. It is assumed that the events are enumerated such that $t_0 \le t_1 \le \cdots \le t_n$ holds. The continuous (auxiliary) variable $r_{ek}$ for $e \in \mathcal{E}, k \in R$ represents the quantity of resource $k$ required immediately after event $e$.

Using these variables, the SEE-formulation was introduced as follows:

$$\min \quad t_n \tag{1}$$

$$\text{s.t.} \quad t_0 = 0 \tag{2}$$

$$t_{e+1} - t_e \ge 0 \quad (e \in \mathcal{E}\backslash\{n\}) \tag{3}$$

$$t_f - t_e - p_i x_{ie} + p_i(1 - y_{if}) \ge 0 \quad (i \in A;\ e,f \in \mathcal{E}, e < f) \tag{4}$$

$$\sum_{e \in \mathcal{E}} x_{ie} = 1 \quad (i \in A) \tag{5}$$

$$\sum_{e \in \mathcal{E}} y_{ie} = 1 \quad (i \in A) \tag{6}$$

$$\sum_{e'=e}^{n} y_{ie'} + \sum_{e'=0}^{e-1} x_{je'} \le 1 \quad ((i,j) \in E;\ e \in \mathcal{E}) \tag{7}$$

$$r_{0k} - \sum_{i \in A} b_{ik} x_{i0} = 0 \quad (k \in R) \tag{8}$$

$$r_{ek} - r_{e-1,k} + \sum_{i \in A} b_{ik}(y_{ie} - x_{ie}) = 0 \quad (e \in \mathcal{E}\backslash\{0\};\ k \in R) \tag{9}$$

$$r_{ek} \le B_k \quad (e \in \mathcal{E}; k \in R) \tag{10}$$

$$x_{ie}, y_{ie} \in \{0,1\} \quad (i \in A;\ e \in \mathcal{E}) \tag{11}$$

$$t_e \ge 0 \quad (e \in \mathcal{E}) \tag{12}$$

$$r_{ek} \ge 0 \quad (e \in \mathcal{E};\ k \in R) \tag{13}$$

Additionally, the following inequalities were integrated taking into account time windows $[ES_i, LS_i]$ for the activities $i \in A$, where $ES_i$ and $LS_i$ denote the earliest and latest starting time for $i$, respectively:

$$ES_i x_{ie} \le t_e \le LS_i x_{ie} + LS_{n+1}(1 - x_{ie}) \quad (i \in A; e \in \mathcal{E}) \tag{14}$$

$$(ES_i + p_i) y_{ie} \le t_e \le (LS_i + p_i) y_{ie} + LS_{n+1}(1 - y_{ie}) \quad (i \in A; e \in \mathcal{E}) \tag{15}$$

$$ES_{n+1} \le t_n \tag{16}$$

The objective function (1) consists in minimizing the completion time $t_n$ of an activity processed last. Constraint (2) indicates that the first event starts at time 0, (3) takes care of the ordering of the events. Inequalities (4) ensure that if $x_{ie} = y_{if} = 1$ (i.e. $i$ starts at event $e$ and completes at event $f$), then $t_f \ge t_e + p_i$ holds. For all other combinations of values for $x_{ie}$ and $y_{if}$ we have either $t_f \ge t_e$ or $t_f \ge t_e - p_i$, which are covered by (3). Constraints (5) and (6) guarantee that each activity starts and ends exactly once. Constraint (7) ensures that the given precedence constraints are respected: If a predecessor $i$ of $j$ ends at event $e$ or later (i.e. $\sum_{e'=e}^{n} y_{ie} = 1$), then $\sum_{e'=0}^{e-1} x_{je'}$ must be zero, i.e. $j$ cannot start before event $e$. Equalities (8) set the initial quantities of each resource $k$ needed immediately after time 0. Equalities (9) describe the recursion for calculating the $r_{ek}$-values for the other events, namely the quantity of resource $k$ needed immediately after time $t_e$ is equal to the quantity of resource $k$ needed immediately after time $t_{e-1}$ plus the quantity of resource $k$ needed by the activities starting at time $t_e$ minus the quantity of resource $k$ needed by the activities completing at time $t_e$. Constraints (10) limit the quantity of resource $k$ needed immediately after time $t_e$ to the availability of resource $k$. Inequalities (14) and (15) ensure that an activity cannot start before its earliest starting time nor after its latest starting time. Furthermore, (16) says that the project cannot end before the earliest start time of the dummy finishing activity $n+1$.

Unfortunately, this formulation is not correct since an activity may end at the same time or before it is started (i.e. we may set $x_{ie} = y_{if} = 1$ for events $f < e$). For simplicity, we consider the formulation without the inequalities (14)–(16) for the time windows since they do not influence correctness.

For a counterexample, consider a project with $n=4$ activities, $m=2$ resources with capacities $B_1 = 5$, $B_2 = 7$, a precedence relation (2,3), and the following data:

| $i$ | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| $p_i$ | 4 | 3 | 5 | 8 |
| $b_{i1}$ | 2 | 1 | 2 | 2 |
| $b_{i2}$ | 3 | 5 | 2 | 4 |

An optimal schedule with makespan $C_{\max} = 12$ is shown in Fig. 1.

Let us consider the solution

$$t_0 = t_1 = t_2 = t_3 = t_4 = 0$$
$$x_{14} = x_{21} = x_{34} = x_{41} = 1$$
$$y_{10} = y_{21} = y_{31} = y_{40} = 1$$
$$r_{41} = 4, \quad r_{12} = r_{22} = r_{32} = 2, \quad r_{42} = 7$$

where all other values are equal to zero. In the following, we show that this solution is feasible for the SEE-formulation (1)–(13): That the constraints (2), (3), (5), (6), and (11)–(13) are satisfied, can be seen immediately. Conditions (4) hold because for all $e < f$ we have $(x_{ie}, y_{if}) \ne (1,1)$ for $i = 1,2,3,4$. Therefore, (4) is equivalent to $t_f \ge t_e$ or $t_f \ge t_e - p_i$, which is satisfied due to (3). Condition (7) is fulfilled because for $(i,j) = (2,3)$ we have $\sum_{e'=1}^{n} y_{2e'} + x_{30} = 1 + 0 = 1$ and $\sum_{e'=e}^{n} y_{2e'} = 0$ for $e > 1$. That (8)–(10) are satisfied can be seen by evaluating (8) and (9). For example, for the

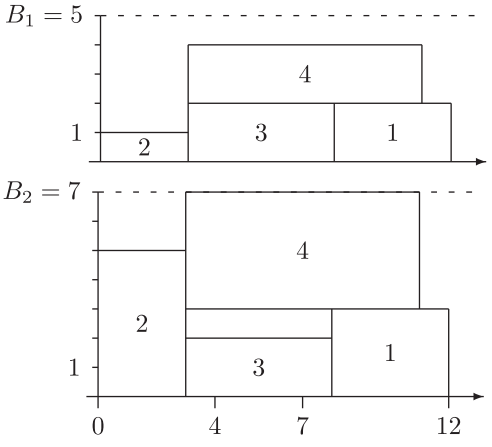


**Fig. 1.** An optimal schedule for the example.