European Journal of Operational Research 221 (2012) 27–37

Contents lists available at SciVerse ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

Mixed-integer linear programming for resource leveling problems

Julia Rieck *, Jürgen Zimmermann, Thorsten Gather

Clausthal University of Technology, Institute of Management and Economics, Operations Research Group, Julius-Albert-Str. 2, 38678 Clausthal-Zellerfeld, Germany

ARTICLE INFO

Article history:
Received 7 July 2011
Accepted 4 March 2012
Available online 16 March 2012

Keywords:
Project scheduling
Resource leveling
Minimum and maximum time lags
Lower and upper bounds
Cutting planes

ABSTRACT

We consider project scheduling problems subject to general temporal constraints, where the utilization of a set of renewable resources has to be smoothed over a prescribed planning horizon. In particular, we consider the classical resource leveling problem, where the variation in resource utilization during project execution is to be minimized, and the so-called "overload problem", where costs are incurred if a given resource-utilization threshold is exceeded. For both problems, we present new mixed-integer linear model formulations and domain-reducing preprocessing techniques. In order to strengthen the models, lower and upper bounds for resource requirements at particular points in time, as well as effective cutting planes, are outlined. We use CPLEX 12.1 to solve medium-scale instances, as well as instances of the well-known test set devised by Kolisch et al. (1999). Instances with up to 50 activities and tight project deadlines are solved to optimality for the first time.

1. Introduction

A project is a unique and temporary endeavor that can be subdivided into various activities that require time and renewable resources, such as machines, equipment, or manpower, for their execution. Usually, projects involve general temporal constraints among activities resulting from technological or organizational restrictions. Project scheduling consists of determining start times for all activities such that temporal and/or resource constraints are satisfied and some objective is optimized (see e.g. Józefowska and Węglarz, 2006).

Resource leveling problems (RLPs) arise whenever it is expedient to reduce the fluctuations in patterns of resource utilizations over time, while maintaining compliance with a prescribed project completion time. In particular, in cases where even slight variations in resource needs represent financial burden or heightened risks of accidents, a resource leveling approach helps to schedule the project activities such that the resource utilization will be as smooth as possible over the entire planning horizon (cf. Demeulemeester and Herroelen, 2002). Under resource leveling, no resource limits are typically imposed. Therefore, only the time lags between individual activities form the project constraints.

Resource leveling has received little attention in the academic literature. A bunch of instances with 30 activities (cf. Kolisch et al., 1999) remain to be solved optimally. In order to compensate for that dearth of research, we consider exact methods for the "classical resource leveling problem", where variations in resource utilizations within the project duration are to be minimized (cf. Burgess and Killebrew, 1962). In addition, we study the "overload problem", where costs are incurred if either a given supply of some renewable resources or a threshold for the resource utilization is exceeded (cf. Easa, 1989). New mixed-integer linear models and domain-reducing preprocessing techniques are devised for both problems. We obtain promising results on the well-known test instances of Kolisch et al. (1999) using CPLEX 12.1. For the first time, all problem instances with 30 activities are solved to optimality with respect to the minimum project duration.

In Section 2, we formally describe the resource leveling problem using two different objective functions and present its mathematical background. In Section 3, we investigate an interesting application of resource leveling that substantiates both the objective functions we have proposed and the structuring of the problem instances we have used in our experimental performance analysis. Section 4 is devoted to a literature review on exact solution methods for resource leveling, where we sketch the most common approaches and present known mathematical model formulations. Based on those models, we proceed to describe methods for linearizing the corresponding objective functions and improving the quality of the resulting formulations in terms of computation time and solution gap (cf. Section 5). The results of a comprehensive performance analysis are given in Section 6. Finally, conclusions are presented in Section 7.

2. Problem description

In the remainder of this paper, we consider projects specified by activity-on-node networks $N = (V, A; \delta)$, where V is the set of vertices and A is the set of arcs with weight δ. Vertex set

* Corresponding author.
E-mail addresses: julia.rieck@tu-clausthal.de (J. Rieck), juergen.zimmermann@tu-clausthal.de (J. Zimmermann).

0377-2217/$ - see front matter
http://dx.doi.org/10.1016/j.ejor.2012.03.003

$V := \{0, 1, \ldots, n, n+1\}$ consists of $n \geqslant 1$ activities, $1, \ldots, n$, that have to be carried out without interruption, and two fictitious activities, 0 and $n+1$, that represent the beginning and completion of the underlying project, respectively. Each activity has to be started no earlier than the project beginning, and must be completed by project termination.

We denote the start time of activity $i \in V$ by S_i and assume that every project begins at time zero, i.e. $S_0 := 0$. Then, S_{n+1} equals the project duration. If activity j cannot be started earlier than $d_{ij}^{min} \in \mathbb{Z}_{\geqslant 0}$ time units after activity i (minimum time lag), i.e. $S_j - S_i \geqslant d_{ij}^{min}$, we introduce an arc $\langle i, j \rangle$ having weight $\delta_{ij} := d_{ij}^{min}$ into network N. In the event that activity j can be begun as soon as activity i has been concluded, i.e. $d_{ij}^{min} = p_i$, the minimum time lag is referred to as a "precedence constraint". If activity j must be started no later than $d_{ij}^{max} \in \mathbb{Z}_{\geqslant 0}$ time units after activity i (maximum time lag), i.e. $S_j - S_i \leqslant d_{ij}^{max}$, we introduce a backward arc $\langle j, i \rangle$ with weight $\delta_{ji} := -d_{ij}^{max}$. The resulting arc set A contains at most $|V||V-1|$ arcs representing the temporal constraints $S_j - S_i \geqslant \delta_{ij}$ among the start times of activities $i, j \in V$.

A sequence of start times $S = (S_0, S_1, \ldots, S_{n+1})$, where $S_i \geqslant 0$, $i \in V$, and $S_0 = 0$, is termed a "schedule". A schedule is said to be feasible if it satisfies all temporal constraints of the project given by minimum and maximum time lags. The set of all feasible schedules is denoted by $\mathcal{S}_T$. Let $\bar{d} \geqslant 0$ be a prescribed maximum project duration (i.e. a project-completion deadline). The problem of finding an optimal (feasible) schedule for some objective function $f : \mathbb{R}_{\geqslant 0}^{n+2} \to \mathbb{R}$ to be minimized may be formulated as follows:

$$\left.\begin{array}{ll} \text{Minimize} & f(S) \\ \text{subject to} & S_j - S_i \geqslant \delta_{ij} \quad \langle i,j \rangle \in A \\ & S_0 = 0 \\ & S_{n+1} \leqslant \bar{d} \\ & S_i \geqslant 0 \quad i \in V. \end{array}\right\} \tag{1}$$

For an activity $i \in V$, inequality $S_i \geqslant 0$ is already implied by $S_0 := 0$ and the assumption that no activity can be started prior to the project beginning. Furthermore, since we assume in the following that network N contains an arc $\langle n+1, 0 \rangle$ having weight $\delta_{n+1,0} := -\bar{d}$ in order to ensure compliance with the prescribed project deadline, inequality $S_{n+1} \leqslant \bar{d}$ also becomes redundant. As has been shown by Bartusch et al. (1988), the feasible region $\mathcal{S}_T$ of problem (1) is non-empty, iff N contains no cycle of positive length, which can be checked in polynomial time (cf. Ahuja et al., 1993, Section 5.5).

Due to the prescribed project deadline, the set of feasible start times of activity $i \in V$ forms a proper time window $[ES_i, LS_i]$, where ES_i is the earliest and LS_i the latest start time of activity i with respect to the given temporal constraints. By definition, $ES_0 = LS_0 := 0$. For a specified activity $i \in V \setminus \{0\}$, both the earliest start time, ES_i, which equals the length of a longest path from node 0 to node i, and the latest start time, LS_i, which equals the negative of the longest path length from node i to node 0, can be determined by applying some label-correcting algorithm (see e.g. Ahuja et al., 1993, Section 5.4). The total float, $TF_i := LS_i - ES_i$, $i \in V$, is the maximum length of time, by which the start of activity i may be delayed beyond its earliest start time, without causing project completion to be delayed beyond the final deadline given by $\bar{d}$. An activity i is termed "critical" if a delay in its start will cause a delay in completing the entire project. The total float is therefore zero for critical activities, and has some positive value for non-critical activities.

Let $\mathcal{R}$ be the set of renewable resources required for carrying out the project activities. Every activity $i \in V$ has a given processing time $p_i \in \mathbb{Z}_{\geqslant 0}$, and requires $r_{ik} \in \mathbb{Z}_{\geqslant 0}$ units of resource $k \in \mathcal{R}$ taken up by processing activity i, commencing with its start time S_i (inclusively), through to its completion time $S_i + p_i$ (exclusively). An activity i is referred to as "event" if $p_i = 0$; otherwise, it is regarded as a real activity. Every real activity i is presumed to be performed during the half-open time interval $[S_i, S_i + p_i[$. In case of the fictitious activities, we set $p_0 = p_{n+1} := 0$ and $r_{0k} = r_{n+1,k} := 0$ for all $k \in \mathcal{R}$. Given some schedule S, the set of (real) activities in progress at time t, which is also termed the "active set", is given by $\mathcal{A}(S,t) := \{i \in V | S_i \leqslant t < S_i + p_i\}$. Thus, $r_k(S,t) := \sum_{i \in \mathcal{A}(S,t)} r_{ik}$ represents the total amount of resource $k \in \mathcal{R}$ required for those activities in progress at time t. The resource profiles $r_k(S, \cdot) : [0, \bar{d}] \to \mathbb{R}_{\geqslant 0}$ are step functions continuous from the right at their jump points.

If the resources necessary to carry out the activities involved should be distributed evenly over the time horizon, we speak of *resource leveling*. Different objective functions are considered in the literature (see e.g. Neumann and Zimmermann, 1999, 2000), depending on how variations in resource utilizations are measured. In what follows, we consider two resource leveling functions having broad areas of application.

In practice, companies often want to realize smooth resource profiles for a given project duration, and aim at penalizing high resource utilizations more than low resource utilizations. Let $c_k \geqslant 0$ be the cost incurred per unit of resource $k \in \mathcal{R}$, and per time unit. The "classical resource leveling objective function" will then be given by

$$f(S) := \sum_{k \in \mathcal{R}} c_k \int_{t \in [0,\bar{d}]} r_k^2(S,t)\, dt. \tag{RL1}$$

(RL1) represents the total squared utilization cost for a given schedule S (cf. Burgess and Killebrew, 1962; Harris, 1990). A possible application can be found in make-to-order manufacturing operations, where an even-workload distribution of resources is required (cf. Ballestin et al., 2007). Moreover, (RL1) may be used for avoiding large deviations from prescribed resource-utilization thresholds $Y_k, k \in \mathcal{R}$, since the conditions

$$\begin{aligned} &\sum_{k \in \mathcal{R}} c_k \int_{t \in [0,\bar{d}]} (r_k(S,t) - Y_k)^2 dt \\ &= \sum_{k \in \mathcal{R}} c_k \left(\int_{t \in [0,\bar{d}]} r_k(S,t)^2 dt - 2\, Y_k \sum_{i \in V} r_{ik}\, p_i + \bar{d}\, Y_k^2 \right) \\ &= \sum_{k \in \mathcal{R}} c_k \int_{t \in [0,\bar{d}]} r_k^2(S,t) dt + K \end{aligned}$$

are satisfied with some $K \in \mathbb{R}$.

Employers are usually required to pay overtime premiums to employees who work more than the standard hours. Additional costs for covering the positive deviations from the desired resource utilizations $Y_k, k \in \mathcal{R}$, will therefore be incurred (cf. Easa, 1989; Bandelloni et al., 1994). In order to take this option into account, we consider the "total overload cost function"

$$f(S) := \sum_{k \in \mathcal{R}} c_k \int_{t \in [0,\bar{d}]} (r_k(S,t) - Y_k)^+ dt. \tag{RL2}$$

In case no thresholds Y_k, e.g. the standard weekly hours, have been prescribed, Y_k may be chosen equal to the (rounded) average resource utilizations, i.e. $Y_k := \sum_{i \in V} \lceil r_{ik}\, p_i / \bar{d} \rceil$.

If time t is discrete, the integrals appearing in (RL1) and (RL2) could be replaced by summations. As has been shown by Neumann et al. (2003), problem (1) is $\mathcal{NP}$-hard in the strong sense in case of both resource leveling variants. However, both objective functions and the set of feasible solutions have nice properties that can be exploited along the way to an optimal solution. Firstly, the feasible region represents a convex polytope of dimension $n+1$ if network N contains neither any redundant time lags nor cycles of length zero. An algorithm for eliminating redundant arcs may be found in Habib et al. (1993) or Gather et al. (2011). The activities of some cycle of length zero will be strictly interlinked and may be replaced by a single node (cf. Neumann et al., 2003). Furthermore, every binding temporal constraint, $S_j = S_i + \delta_{ij}$, $i, j \in V$, defines a facet of the feasible region (cf. Hagmayer, 2006). Secondly, objective