Expert Systems With Applications 97 (2018) 193–204

Contents lists available at ScienceDirect

Expert Systems With Applications

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

# Multi-mode resource leveling in projects with mode-dependent generalized precedence relations

Hongbo Li, Xuebing Dong*

*School of Management, Shanghai University, Shangda Road 99, Shanghai 200444, China*

ARTICLE INFO

*Article history:*
Received 7 September 2017
Revised 15 December 2017
Accepted 16 December 2017
Available online 16 December 2017

*Keywords:*
Project management
Project scheduling
Resource leveling
Generalized precedence relations
Multi-mode case
Estimation of distribution algorithm

ABSTRACT

In real-life project management, resource leveling is an important technique to ensure the effective use of resources, in which activities (a) can often be executed in alternative modes and (b) are constrained by precedence relations with minimum and maximum time lags that can be modeled using generalized precedence relations (GPRs). In addition, the values of the time lags tend to depend on activity modes. The resource leveling problem with multiple modes and mode-dependent GPRs (MRLP-GPR) is a generalization of the classic NP-hard resource leveling problem. To our knowledge, no literature exists regarding the MRLP-GPR.

We propose several heuristics for the MRLP-GPR built upon two solution approaches: (a) a steepest descent algorithm and a fast descent algorithm that are based on a decomposition approach and (b) a hybrid estimation of distribution algorithm (EDA), which is based on an integration approach. Extensive computational experiments on a large number of benchmark instances are conducted to evaluate the proposed heuristics. A comparison of the results shows that our EDA outperforms or is competitive with three baseline heuristics (a random search algorithm and two variants of a genetic algorithm that is the best-performing metaheuristic for the single-mode resource leveling problem with GPRs). Our results can serve as a benchmark for future research. Our model and solution algorithms provide an automatic tool for the project manager's multi-mode resource leveling decision-making.



## 1. Introduction

For projects in practice, resource leveling is an important technique to ensure the effective use of resources. To a large extent, many negative effects can be avoided by resource leveling, such as hastily allocating temporary resources, frequently hiring and dismissing employees, and idle resources caused by reserving a high security level (Li, Xiong, Liu, & Li, 2017). This condition motivates a study of the resource leveling problem (RLP) that aims to minimize the resource utilization fluctuations through scheduling activities under precedence constraints and the project deadline constraint (Li & Demeulemeester, 2016; Li, Xu, & Demeulemeester, 2015; Markou, Koulinas, & Vavatsikos, 2017; Neumann, Schwindt, & Zimmermann, 2003). Here, we propose three heuristics for the RLP with multiple modes and mode-dependent generalized precedence relations (GPRs).

In real-life project management, activities can often be executed in alternative modes, indicating that activity duration can be changed by varying the amount of resources allocated (Heilmann, 2001). However, few studies have considered multiple modes in the RLP. Coughlan, Lübbecke, and Schulz (2015) developed a branch-price-and-cut algorithm for the multi-mode RLP (MRLP) with the objective of minimizing the resource availability cost. However, their problem setting is relatively simplistic; for example, they assume that multiple resource types exist but that each activity requires only one resource type to execute. Behrouz, Hojjat, and Esmaeil (2012) designed a branch-and-bound procedure for the MRLP. Based on constraint programming, Menesi and Hegazy (2015) proposed a bi-objective multi-mode project scheduling model in which one of the objectives is to minimize the peak resource demand. Guo, Li, Zhang, and Ye (2012) presented a particle swarm optimization algorithm for a multi-objective multi-mode project scheduling problem. One of their objective functions was to minimize the resource usage variation.

The above-mentioned studies on the MRLP focus only on critical path method (CPM)-type precedence relations, i.e., the finish-start precedence relations with zero time lag. To our knowledge, a more generalized precedence relation (also known as minimum and maximum time lags) has not been considered in the MRLP. The advantage of the GPR is that it can easily describe many com-

* Corresponding author.
*E-mail address:* dongxuebing116@sina.com (X. Dong).

https://doi.org/10.1016/j.eswa.2017.12.030
0957-4174/

plex precedence constraints, such as the project/activity due date, parallel execution of activities, time window constraints, and so on (Dorndorf, Pesch, & Phan-Huy, 2000).

Currently, the resource leveling literature that considers GPR focuses only on solving the single-mode RLP-GPR. For exact methods, a branch-and-bound algorithm (Gather, Zimmermann, & Bartels, 2011; Neumann & Zimmermann, 2000) and a mixed-integer programming algorithm (Kreter, Rieck, & Zimmermann, 2014; Rieck & Zimmermann, 2015; Rieck, Zimmermann, & Gather, 2012) have been proposed to handle the RLP-GPR. For heuristic approaches, Neumann and Zimmermann (1999, 2000) have devised polynomial heuristics and priority-rule-based heuristics for the RLP-GPR. Metaheuristics based on a tabu search algorithm (Neumann & Zimmermann, 2000), an iterated greedy algorithm (Ballestín, Schwindt, & Zimmermann, 2007), and a genetic algorithm (Li et al., 2017) have also been developed.

When multiple modes and GPRs come into play, the time lags tend to become mode-dependent (De Reyck & Herroelen, 1999). Mode-dependent time lags mean that the time lag between any two activities depends on the modes (durations) of both activities. For example, let us consider two activities in a software project: "testing module A" (activity 1) and its successor "testing module B" (activity 2). Assume that there are two modes for activity 1: module A can be tested by two software engineers with one hour (mode 1) or by one software engineer with two hours (mode 2). Activity 2 has one mode. Since activity 2 is the successor of activity 1, it can be started as soon as activity 1 is finished. Therefore, if activity 1 is executed in mode 1, activity 2 can be started one hour after the start of activity 1, implying that the time lag between the start of activities 1 and 2 is one hour. Similarly, if activity 1 is executed in mode 2, then the time lag between the start of activities 1 and 2 is two hours. Mode-dependent time lags have been considered in some studies on the multi-mode resource-constrained project scheduling problem with generalized precedence relations (MRCPSP-GPR), and exact and heuristic methods have been developed (Ballestín, Barrios, & Valls, 2013; Barrios, Ballestín, & Valls, 2011; Heilmann, 2001, 2003; Jędrzejowicz & Ratajczak-Ropel, 2011).

However, to our knowledge, no literature exists on the RLP with multiple modes and mode-dependent generalized precedence relations (MRLP-GPR). MRLP-GPR is a generalization of RLPs that have been proven to be NP-hard (Neumann et al., 2003). Specifically, MRLP-GPR is a generalization of the MRLP in which only CPM-type precedence relations are considered. In addition, MRLP-GPR is a generalization of the RLP-GPR in which each activity has only one mode.

In this paper, the MRLP-GPR is studied for the first time and efficient and effective heuristics are developed. The following novel features and design ensure the efficiency and effectiveness of the proposed heuristics:

(a) By analyzing the character of the MRLP-GPR, we propose two solution approaches: decomposition and integration. These approaches reflect how we deal with two sub-problems (a mode assignment problem and the RLP-GPR) of the MRLP-GPR.
(b) Based on the decomposition approach, we devise a steepest descent algorithm and a fast descent algorithm. Both of the descent algorithms are equipped with a relatively fast and simple priority-based heuristic. This heuristic has a computational complexity of $O(n^2)$ and has the potential to find optimal solutions. This heuristic can also be adapted for the so-called multi-mode resource leveling schedule generation scheme.
(c) Following the integration approach, we develop a hybrid estimation of distribution algorithm (EDA) that is mixed with genetic algorithm (GA) operators. In our EDA, new individuals are generated jointly using the problem-specific probability model of the EDA and the crossover and mutation operators of the GA. Our EDA is also enhanced by certain new characteristics, such as multi-mode schedule encoding and decoding mechanisms, a problem-specific probability model updating mechanism, and a probability-generating mechanism (PGM).

To evaluate and compare the proposed heuristics, we perform extensive computational experiments on a large number of benchmark instances. The computational results show that our EDA outperforms or is competitive with the baseline heuristics (a random search algorithm and two variants of a GA (Li et al., 2017) that is the best-performing metaheuristic for the RLP-GPR). Our results can serve as a benchmark for future research. In addition, our algorithms can be easily embedded into expert systems or project management software. When facing real-world multi-mode resource leveling problems, the project manager can choose satisfactory schedules with the help of our algorithms. This will make our algorithms more practical.

The remainder of this paper is organized as follows. We first introduce the MRLP-GPR in the next section. In Section 3, we describe the decomposition and integration solution approaches for the MRLP-GPR. Two descent algorithms based on the decomposition approach are presented in Section 4. A hybrid EDA based on the integration approach is proposed in Section 5. Section 6 presents the computational results and comparisons. The last section concludes the paper.

## 2. The MRLP-GPR

### 2.1. Problem description

The MRLP-GPR can be stated as follows. We use a weighted cyclic activity-on-node network $G=(N, A)$ to represent a project. $N$ is the set of nodes and indicates the activities, $N=\{0, 1, \ldots, n, n+1\}$. $A$ is the set of directed arcs and denotes the GPRs, and $A \subseteq N \times N$. The activities are numbered from 0 to $n+1$. Dummy activities 0 and $n+1$ indicate the start and the end of the project, respectively.

Allocating different amounts of resources to each activity leads to different activity durations. Thus, each non-dummy activity $i$ can be executed in one of $M_i$ modes. Each mode $m_i$ ($m_i \in \{1, \ldots, M_i\}$) represents an alternative combination of activity duration and resource requirements. There are $K$ resource types. When executed in mode $m_i$, each non-dummy activity $i$ has an integer duration $d_{im_i}$ and requires $r_{im_ik}$ renewable resources for the resource type $k$ per time period ($k=1, 2, \ldots, K$). An activity cannot be preempted once it has been started.

Given an activity $i$, $s_i$ represents its starting time. Thus, the starting time of the dummy end activity $s_{n+1}$ denotes the project's finish time. The project's deadline $\bar{d}$ is predefined, i.e., $s_{n+1} \leq \bar{d}$. Given the activity mode assignments and starting times, let $u_{kt}$ denote the resource usage for the resource type $k$ during a time period $t$, with $t = 1, 2, \ldots, \bar{d}$. $u_{kt}$ can be calculated as $u_{kt} = \sum_{i \in ACT_t} r_{im_ik}$, where $ACT_t$ is the set of activities that are being executed in the time period $t$, $ACT_t = \{i | s_i < t \leq s_i + d_{im_i}\}$.

The activities are subject to GPRs, which refers to the minimum and maximum lag between the starting times of the activities (Neumann et al., 2003). Because start-to-completion, completion-to-start and completion-to-completion time lags can be converted into start-to-start time lags (Bartusch et al. 1988), it is sufficient to consider only the minimum and maximum lag between the starting time of the activities. For activities $i$ and $j$ that are executed in modes $m_i \in \{1, \ldots, M_i\}$ and $m_j \in \{1, \ldots, M_j\}$, respectively, we associate each arc $\langle i, j \rangle \in A$ with a weight $\delta_{ij}^{m_im_j}$ representing the time lag between the starting times of the activities $i$ and $j$, i.e., the start-to-start time lag. Note that time lags are mode-dependent,