Expert Systems With Applications 57 (2016) 91–103

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Priority-based heuristics for the multi-skill resource constrained project scheduling problem

Bernardo F. Almeida[a,*], Isabel Correia[b], Francisco Saldanha-da-Gama[a]

[a] *Departamento de Estatística e Investigação Operacional / Centro de Matemática e Aplicações Fundamentais — Centro de Investigação Operacional, Faculdade de Ciências, Universidade Lisboa, Lisboa 1749-016, Portugal*

[b] *Departamento de Matemática / Centro de Matemática e Aplicações, Faculdade de Ciências e Tecnologia, Universidade Nova Lisboa, Caparica 2829-516, Portugal*

ARTICLE INFO

Article history:
Received 3 October 2015
Revised 5 March 2016
Accepted 6 March 2016
Available online 16 March 2016

Keywords:
Project scheduling
Multi-skilled resources
Parallel scheduling heuristic
Activity grouping
Resource weight

ABSTRACT

In this paper we investigate one of the most recent extensions of the Resource Constrained Project Scheduling Problem (RCPSP): the Multi-Skill Resource Constrained Project Scheduling Problem (MSRCPSP). For this complex problem we propose the use of a parallel scheduling scheme. Such scheme has been successfully applied to the RCPSP. Nevertheless, in order to apply it to the MSRCPSP two new concepts are developed: resource weight and activity grouping. We discuss such concepts and use them for the new heuristic framework proposed. A series of computational tests performed using a large number of instances and reported in this paper shows that the new heuristic is very effective in finding high quality solutions within very small CPU times.

1. Introduction

Project scheduling is one of the most important topics within project management. It consists of finding a schedule for the activities underlying a project such that a predefined performance measure is optimized but without violating any existing constraint (e.g., precedence or resource constraints). In the recent decades these problems have gained much prominence since many companies and organizations often have to deal with projects that are increasingly larger and more complex.

The basic version of a project scheduling problem consists of a set of non-preemptive activities linked by precedence relations. The most common objective is to minimize the makespan of the project by scheduling all activities without violating precedence constraints.

One of the most well-known extensions of this problem is the Resource Constrained Project Scheduling Problem (RCPSP). In this case, the execution of an activity requires specific quantities of renewable resources that are available in limited amounts. Many variants of the RCPSP have been studied in the literature (the reader can refer to the surveys by Hartmann and Briskorn 2010, and Weglarz, Józefowska, Mika, and Waligóra 2011, and to the references therein).

In the RCPSP each resource has a single capability, i.e., masters a single skill. However, in many real applications resources are multi-skilled, that is, each resource masters one or several skills. Within this context each activity may require several resources per each skill needed for its execution. This extension of the RCPSP is known as the Multi-Skill Resource Constrained Project Scheduling Problem (MSRCPSP). This type of problem easily emerges when human resources or multi-purpose machines are involved in a project.

In the MSRCPSP not only is it necessary to decide the specific resources that will be assigned to each activity but also the skills with which they will contribute. This represents one extra type of decision in comparison with the classical RCPSP which, as made clear by Correia and Saldanha-Gama (2015), makes the problem more challenging.

To the best of the authors' knowledge, Néron (2002) represents the first contribution to the study of the MSRCPSP. In that work, two lower bounds are proposed: one is based on linear programming and the other one on energetic reasoning. The former is an adaptation of a linear programming approach presented by Carlier and Néron (2003) for the RCPSP; the latter has been previously applied by Baptiste, Pape, and Nuijten (1999) to the cumulative scheduling problem, a particular case of the RCPSP. Bellenguez-Morineau and Néron (2005), (2007) and Bellenguez-Morineau (2008) study a multi-skill project scheduling problem

* Corresponding author. Tel.: +351963753724.
E-mail addresses: bernardo.almeida@alunos.fc.ul.pt (B.F. Almeida), isc@fct.unl.pt (I. Correia), fsgama@fc.ul.pt (F. Saldanha-da-Gama).

http://dx.doi.org/10.1016/j.eswa.2016.03.017

with unavailability periods associated with the resources. The goal is to minimize the makespan of the project. Bellenguez-Morineau and Néron (2005) consider hierarchical levels for the skills, i.e., skills can be performed with different efficiencies according to the assigned resources. These resources are referred to in the literature as resources with heterogeneous efficiencies, and have also been considered, among others, by Heimerl and Kolisch (2010). For each activity, the input of the problem includes the processing time, the required skills and the execution levels for those skills. More than one resource may be necessary for each skill-level combination.

Li and Womer (2009) consider a multi-skill resource constrained project scheduling problem where each activity requires only one resource per each skill needed for its execution. The goal is to minimize the total cost associated with the resources satisfying some predefined deadline for the project to be completed. Correia, Lourenço, and Saldanha-da-Gama (2012) present a mixed-integer linear programming (MILP) formulation for the MSRCPSP. The authors enhanced that model with several sets of additional inequalities which aim at restricting the search space by identifying the optimal values of some binary variables. The goal is again to minimize the makespan of the project. Correia and Saldanha-da-Gama (2014) study the impact of fixed and variable costs associated with the resources in a MSRCPSP. Recently Correia and Saldanha-Gama (2015) discuss modeling issues and present a general modeling framework for project scheduling and staffing problems.

A related problem was investigated by Fırat and Hurkens (2012). The authors consider a mixed-integer based approach for a multi-skill work loading problem. Different levels of efficiency are considered for each skill. Each activity has requirements for each skill-level combination. The goal is to determine the workloads of teams so that the number of tasks processed in each workday is maximized.

In some multi-project scheduling problems, multi-skilled resources have also been considered but in a different setting. For example, no sequence decisions are to be made. The papers by Gutjahr, Katzensteiner, Reiter, Stummer, and Denk (2008) and Heimerl and Kolisch (2010) are examples of works in this direction.

Since the MSRCPCP is $\mathcal{NP}$-Hard (Correia et al. 2012) the possibility of solving the problem to optimality using exact solution procedures is limited by the size of the instances. However, many real instances of project scheduling problems are quite large. Therefore, having efficient heuristics for finding good quality solutions is of great relevance. This is what we investigate in this paper. In particular, we propose priority-based heuristics that extend to our problem some well-known procedures for the RCPSP.

The remainder of this paper is organized as follows. In Section 2 we detail the MSRCPSP. Afterwards, in Section 3, the new heuristic approach is introduced. The computational experiments performed are presented in Section 4. Finally, in Section 5, the main conclusions are presented and some directions for future research are discussed.

2. Problem description

We consider the Multi-Skill Resource Constrained Project Scheduling problem (MSRCPSP) studied by Correia et al. (2012). This problem consists of a single project that we assume to be represented by an activity-on-node (AON) network $G = (V, E)$ where $V = \{0, 1, \ldots, i, j, \ldots, n+1\}$ denotes the set of activities and E represents the precedence relations between them. Activities 0 and $n+1$ are dummy and represent the beginning and the end of the project, respectively. The precedence relations impose that each activity starts only after all its predecessors have been executed.

In the above acyclic network the weight of an arc $(i, j) \in E$ is equal to p_i—the processing time of activity i. Preemption is not allowed. We assume that the execution of the activities requires resources with different skills. We denote by $\mathcal{R} = \{1, \ldots, k, \ldots, K\}$ the available pool of resources and by $\mathcal{L} = \{1, \ldots, l, \ldots, L\}$ the set of all skills involved in the project.

In addition to the precedence constraints, we consider resource constraints that in the case of the MSRCPSP are rather more complex than those found in the classical RCPSP. In our problem, the resources are renewable and each resource $k \in \mathcal{R}$ masters one or several skills that we gather in a set denoted by $\mathcal{L}^k$. Furthermore, each activity $i \in V$ requires a set of skills in order to be executed. We gather such skills in a set denoted by $\mathcal{L}_i$. Each activity $i \in V$ may require more than one resource mastering each skill needed for its execution. We denote by r_{il} the number of resources mastering skill $l \in \mathcal{L}_i$ required by activity $i \in V$.

The goal of the MSRCPSP is to decide for each activity, its starting time as well as the resources that should be allocated to it in order to minimize the makespan, i.e., in order to minimize the completion time of the entire project.

In the particular problem we are investigating, we make the following assumptions:

- Each resource can contribute with at most one skill it masters to an activity which requires it at a time.
- The allocation of some resource to an activity lasts for the entire duration of the activity.
- The dummy activities have no resource requirements and have a null processing time.
- Parameters p_i $(i \in V \setminus \{0, n+1\})$ and r_{il} $(i \in V \setminus \{0, n+1\}; l \in \mathcal{L}_i)$ are positive integers.

Given our assumptions, we conclude that the makespan is always positive and less than or equal to $\Sigma_{i \in V} p_i$ (makespan for a full sequential execution of all activities).

As it was already mentioned before, this problem is $\mathcal{NP}$-hard (Correia et al. 2012). Therefore, it is important to have some efficient tools for finding good quality feasible solutions in acceptable computational times. In the next section we propose a new heuristic for the presented problem.

3. Heuristic approach

One of the most popular constructive heuristics for the RCPSP is the Parallel Scheduling Scheme (PSS) (Brooks & White 1965). This is an iterative algorithm whose maximum number of iterations is equal to the number of activities in the project. In each iteration and for an appropriate moment in time, a decision set is considered. This set contains all the activities that are candidate to be scheduled, i.e., the activities whose predecessors have had their execution finished, and for which there are available resources to process them. The inclusion of activities in the decision set is performed using the so-called activity *priority values*. These values are assigned to the activities by means of an activity priority rule that, in turn, is a criterion defined for ranking the activities.

The PSS starts by setting a time counter t to zero. Throughout the execution of the algorithm, t will move forward by taking values that depend on the completion times of already scheduled activities. For each value of this counter, a decision set must be built. The overall procedure can be briefly summarized as follows:

While there are activities in the current decision set remaining to be scheduled, the highest ranked activity is selected. That activity is then scheduled to start at time t and the necessary resources are allocated to it. When the decision set at time t is empty or although not empty it is not possible to schedule more activities belonging to this set due to resource constraints, the time counter t is incremented to the minimum completion time of all activities already scheduled; the decision set is then updated, and a new