Computers and Operations Research 93 (2018) 135–150

Contents lists available at ScienceDirect

Computers and Operations Research

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

# An exact composite lower bound strategy for the resource-constrained project scheduling problem

José Coelho[a,b], Mario Vanhoucke[a,c,d,*]

[a] Ghent University, Department of Management information, operations management and technology, Tweekerkenstraat 2, OOST-VLAANDEREN Gent, Belgium
[b] INESC - Technology and Science, Porto (Portugal) and Universidade Aberta, Rua da Escola Politécnica, 147, Lisbon 1269-001, Portugal
[c] Vlerick Business School, Reep 1, Gent 9000, Belgium
[d] UCL School of Management, University College London, 1 Canada Square, London E14 5AA, United Kingdom

ARTICLE INFO

*Article history:*
Received 26 July 2017
Revised 21 December 2017
Accepted 26 January 2018
Available online 2 February 2018

*Keywords:*
Resource-constrained project scheduling
Branch-and-bound
Lower bounds

ABSTRACT

This paper reports on results for the well-known resource-constrained project scheduling problem. A branch-and-bound procedure is developed that takes into account all best performing components from literature, varying branching schemes and search strategies, using the best performing dominance rules and assembling these components into a unified search algorithm. A composite lower bound strategy that statically and dynamically selects the best performing bounds from literature is used to find optimal solutions within reasonable times. An extensive computational experiment is set up to determine the best combination of the various components used in the procedure, in order to benchmark the current existing knowledge on four different datasets from the literature. By varying the network topology, resource scarceness and the size of the projects, the computational experiments are carried out on a diverse set of projects. The procedure was able to find some new lower bounds and optimal solutions for the PSPLIB instances. Moreover, new best known results are reported for other, more diverse datasets that can be used in future research studies. The experiments revealed that even project instances with 30 activities cannot be solved to optimality when the topological structure is varied.



## 1. Introduction

Resource-constrained project scheduling is a widely discussed project management topic which has roots in and relevance for both academic and practical oriented environments. Due to its inherent problem complexity, it has been the subject of numerous research projects leading to a wide and diverse set of procedures and algorithms to construct resource feasible project schedules. Thanks to its practical value, many researchers have extended the basic problem to new problem formulations by taking practical needs into account, resulting in an overwhelming amount of diversified problem types and case-specific algorithmic procedures (cf. e.g. Hartmann and Briskorn, 2010 for an overview of recent extensions).

In this paper, the well-known resource-constrained project scheduling problem (RCPSP) will be solved by a variety of exact branch-and-bound procedures using different lower bounds, branching strategies and dominance rules from literature. This problem type assumes the presence of renewable resources under a limited availability, and makes use of networks with only finish-start relations with minimal time-lags of zero. The scheduling objective is to minimize the total project duration, known as the project makespan. Due to the long history of research for this challenging problem (the problem is known to be strongly NP hard Blazewicz et al., 1983), various solution methods and databases with solution instances have been reported in literature (Vanhoucke et al., 2016), which will be used in the current research to search for improved lower bound values, better heuristic and more optimal solutions for a wide variety of problem instances with different network and resource structures.

The contribution of this paper is fourfold. First, the paper presents a combined lower bound strategy that will be dynamically updated in a search procedure, in order to optimize the trade-off between speed of calculations and quality of the obtained bounds. Secondly, this lower bound strategy is implemented in a branch-and-bound procedure that collects various branching schemes and search strategies as well as dominance rules from literature. In doing so, it aims at providing a very efficient algorithm that solves more instances to optimality in a reasonable time. Thirdly, the best performing components have been assembled in a combined search strategy, leading to new lower bounds for unsolved prob-

* Corresponding author.
*E-mail address:* mario.vanhoucke@ugent.be (M. Vanhoucke).

https://doi.org/10.1016/j.cor.2018.01.017

lems. Finally, the paper also presents best-known solutions for the RCPSP, not only for the well-known standard sets available in literature, but also for less well-known, but fundamentally different datasets.

The outline of this paper can be summarized along the following lines. Section 2 describes the problem statement and briefly reviews the literature relevant to this paper. It describes the most well-known lower bounds and gives an overview of the summary papers available in literature about the RCPSP. Moreover, it describes how we have combined current knowledge from literature into an integrated solution approach. Section 3 shows how the computational experiment has been set up and presents solutions in various ways. It analyses the use of lower bounds to generate a lower bound strategy that will be used in various implementations of a branch-and-bound procedure. Moreover, it assembles the best performing components into a composite search strategy. Section 4 draws general conclusions and highlights paths for future research.

## 2. Problem statement

The resource-constrained project scheduling problem (RCPSP), denoted as $m, 1|cpm|C_{max}$ using the classification scheme of Herroelen et al. (1999) or as $PS|prec|C_{max}$ using the scheme of Brucker et al. (1999) can be stated as follows. A set of activities $N$, numbered from 0 to $n$ ($|N| = n + 1$), is to be scheduled without pre-emption on a set $R$ of renewable resource types. Each activity $i$ has a deterministic duration $d_i$ and requires $r_{ik}$ units of resource type $k = 1, \dots R$, which has a constant availability $a_k$ throughout the project horizon. We assume that $r_{ik} \le a_k$, $i \in N$, $k \in R$. The dummy start and end activities 0 and $n$ have zero duration and do not make use of the renewable resources, while the other activities have a non-zero duration and a non-negative resource requirement. The set $A$ is used to refer to the set of pairs of activities between which a finish-start precedence relationship with time lag 0 exists. We assume graph $G(N, A)$ to be acyclic. A schedule $S$ is defined by an $(n + 1)$-vector of start times $s = (s_0, \dots, s_n)$ which implies an $(n + 1)$-vector of finish times $f$ ($f_i = s_i + d_i$). A schedule is said to be feasible if the precedence and resource constraints are satisfied. The objective of the RCPSP is to find a feasible schedule such that the schedule makespan is minimised.

The research on the RCPSP has been investigated extensively in the past decades of the previous century, and a full literature overview will not be given in this paper. In the mid to end of the 90's, the development of (mainly) exact procedures have reached a peak and resulted in reviews written by Brucker et al. (1999), Herroelen et al. (1998), Icmeli et al. (1993), Kolisch and Padman (2001) and Özdamar and Ulusoy (1995). Due to the need for solving larger project instances and the increasing power of computers, the research focus has shifted in the beginning of the 20th century from exact procedures to the development of meta-heuristics. Kolisch and Hartmann (1999) present a classification and performance evaluation of existing heuristic procedures. An excellent review paper by Kolisch and Hartmann (2006) discusses the different meta-heuristics for the RCPSP and is an update of the previously published paper of Hartmann and Kolisch (2000). More recently, Hartmann and Briskorn (2010) have studied various extensions of the RCPSP while Abdolshah (2014) has given an extensive overview of approaches and solutions for the RCPSP.

In this paper, different lower bounds will be used in a branch-and-bound procedure to solve the RCPSP. This paper has however no intention to provide a full literature overview of the papers on lower bounds and branch-and-bound procedures for the RCPSP. The main inspiration to implement various lower bounds into different proven branch-and-bound procedures comes from two papers in the literature. The first one has been written more than a decade ago. Klein and Scholl (1999) already mentioned at that time that clever branching strategies had been developed, but the lack of strong lower bounds prevented the exact algorithms to solve large-sized instances. The algorithms were, in still are, therefore restricted to medium-sized problem instances with up to 30 activities. A second more recent paper has been written by Moumene and Ferland (2008) who stipulates that the exact methods are essentially implementations of the branch-and-bound technique, but can still only solve small problems having less than 60 activities. In this paper, we investigate why some problems are still unsolvable and aim for better results by combining the building blocks of the currently best-known performing algorithms in a single procedure.

The next sections explain how the branch-and-bound procedure to solve the RCPSP has been developed using an integration between the best lower bound and search components known in literature. In Section 2.1, the lower bounds used in this study are reviewed and the general outline of the way they have been integrated is explained. The specific details of the implementation is given in a computational experiment section later in this paper. Section 2.2 reviews the different components that are known to perform well and that have been embedded in the search procedure of this paper.

### 2.1. Lower bound strategies

While implementing a search for optimal solution using a branch-and-bound procedure that relies on the best-performing components from literature, a selection of fast and good lower bound calculations have been added into a composite lower bound strategy, that contains 15 different implementations using 7 constructive lower bounds and 3 destructive lower bounds from literature, as summarized along the following lines. Table 1 gives an overview of all lower bounds used in this study. Most of the lower bounds have been discussed in the study by Klein and Scholl (1999). In the table, we introduced two abbreviations for each lower bound. The first set of abbreviations, shown in column CV, are the ones that will be used throughout this paper. The second set of abbreviations, shown in the KS column, are the abbreviations originally used in the paper by Klein and Scholl (1999). Since some of the lower bounds have been slightly adapted, the column labelled as "Remark" described the specific implementation of the lower bound different to the original KS proposal. If nothing is mentioned in this column, the lower bound is implemented as described in the original KS paper. Note that the destructive lower bounds had not received any abbreviated name in this study, and is displayed as - in the table. These bounds are nevertheless identical to the reduction rules proposed in the KS paper. A short description of each lower bound is given along the following lines.

- The *critical path based lower bound* ($LB_{cp}$) is the easiest and most widely used lower bound that efficiently calculates the shortest project duration using only the project network data but ignoring the resource constraints. The use of this bound is standard in the project scheduling literature, and therefore, all results will be compared relative to the performance of the algorithm with the critical path based lower bound as a default calculation.
- The *resource capacity lower bound* ($LB_{rc}$) is originally proposed by Patterson and Huber (1974) and simply takes the total work content into account, but completely ignores the network structure of the project instance. Its calculation boils down to dividing the total work content for each renewable resource, obtained by multiplying the resource requirement by the activ-