Omega 71 (2017) 66–84

Contents lists available at ScienceDirect

Omega

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

An adjustable robust optimization model for the resource-constrained project scheduling problem with uncertain activity durations ☆

M.E. Bruni, L. Di Puglia Pugliese *, P. Beraldi, F. Guerriero

Department of Mechanical, Energy and Management Engineering, University of Calabria, 87036 Rende, Italy

ARTICLE INFO

Article history:
Received 6 March 2016
Accepted 20 September 2016
Available online 25 September 2016

Keywords:
Project scheduling
Resource constraints
Robust optimization
Benders decomposition

ABSTRACT

This paper addresses the resource-constrained project scheduling problem with uncertain activity durations. An adaptive robust optimization model is proposed to derive the resource allocation decisions that minimize the worst-case makespan, under general polyhedral uncertainty sets. The properties of the model are analyzed, assuming that the activity durations are subject to interval uncertainty where the level of robustness is controlled by a protection factor related to the risk aversion of the decision maker. A general decomposition approach is proposed to solve the robust counterpart of the resource-constrained project scheduling problem, further tailored to address the uncertainty set with the protection factor. An extensive computational study is presented on benchmark instances adapted from the PSPLIB.

1. Introduction

The resource-constrained project scheduling problem (RCPSP) consists in sequencing and scheduling project activities usually related by precedence and resource constraints involving renewable scarce resources. As comprehensively investigated in the literature, the RCPSP is an outstanding and challenging problem both in practice, since it arises in many important application fields (construction industry [20,40], rolling ingots production [55,57], to mention a few), and in theory.

The majority of the scientific contributions considers the model parameters deterministic [41]. However, in the last years a frank acknowledgment of the uncertainties characterizing the project environment and a growing attention on the project execution have highlighted the need of incorporating the uncertainty in problem parameters as an inevitable feature of the decision-making process [2,52]. In fact, due to employees' absenteeism, delays in materials supply, bad weather conditions and many other uncontrollable factors, some project activities may last longer than expected, threatening the operational viability of the planned schedule.

To address these challenges, a flourishing stream of literature has focused on the RCPSP under uncertainty, where the activity durations are assumed uncertain [21,22]. Two different approaches can be used depending on the genuine interpretation of the RCPSP under uncertainty and the way this uncertainty is tackled. The first approach assumes that uncertainty is represented by random variables with known distribution functions and interprets the RCPSP as a stochastic dynamic optimization problem, where decisions are made each time new information becomes available.

The second approach has mainly dealt with the development of effective and efficient proactive and reactive scheduling procedures. Proactive scheduling aims at generating baseline schedules that incorporate some protection against possible disruptions, whereas reactive scheduling procedures can be invoked during the execution of the project, to repair the baseline schedule by deviating as little as possible from the original one.

Scant attention has been devoted to robust optimization approaches for the RCPSP. The robust optimization methodology [15,16,31] was first introduced for linear programming problems [13] and then extended for mixed-integer linear programming problems. The approach produces solutions that are feasible for all the realizations of the parameters lying within the uncertainty set. The success of this paradigm, in a broad variety of application areas, is mainly due to the fact that this approach is the only reasonable alternative when the distributional information is not readily available. Moreover, it is easy to understand intuitively and it yields computational tractable mathematical programming problems. Originally designed to handle static problems with uncertain parameters, robust optimization was recently extended into a dynamic setting [14,23]. In particular, part of the variables must be determined before the realization of the uncertainty, whilst other variables can be adjusted to the realization of the uncertain parameters, hence offering increased flexibility. This framework overcomes the conservativeness of early static robust approaches producing significantly less conservative solutions than the static case and yielding better objective values.

☆This manuscript was processed by Associate Editor Kimms.

* Corresponding author.
E-mail address: luigi.dipugliapugliese@unical.it (L. Di Puglia Pugliese).

http://dx.doi.org/10.1016/j.omega.2016.09.009
0305-0483/

Following this stream, this paper presents an adjustable robust formulation for the RCPSP where in the first stage, sequencing decisions are taken, concerning the order of the project activities. In the second stage, scheduling decisions are made, allowing the activity starting times to depend on the activity delays realized.

The contributions of this work are manifold. A novel robust optimization model for the RCPSP under polyhedral duration uncertainty is presented. A tailored solution approach equipped with problem-specific cuts and lower bound inequalities is provided. Further, a polynomially solvable case is identified with a specific uncertainty structure, which provides an intuitive interpretation for decision makers who might flexibly adjust the level of risk aversion. The resulting problem is then solved with an enhanced approach that exploits the specific structure of the uncertainty set. The results, collected on instances adapted from the PSPLIB, show that the behavior of the proposed solution approach is strongly related to the characteristics of the instances and that not all the cases can be solved within the time limit of 20 min. In particular, 15 out of the 48 problem classes of the PSPLIB are computationally demanding, for any risk aversion level. The average computational effort, for the solved instances is, on average, around 173.35 s whereas the gap for the unsolved instances is around 38%. The results confirm that robust variants of the RCPSP are computationally demanding, in line with the results presented in [4].

The remainder of the paper is organized as follows. First, we survey the literature in Section 2. A formal definition of the robust RCPSP is given in Section 3. A tailored decomposition approach is presented in Section 4, whereas Section 5 focuses on a specific uncertain set, where the activity durations are subject to interval uncertainty and the level of robustness is controlled by a protection factor. Section 6 discusses the computational results obtained on a set of benchmark problems. Finally, some conclusions are drawn in Section 7. A detailed accounting of the numerical results is given in Appendix A. Formal proofs of theorems are reported in Appendix B.

2. Related literature

The RCPSP has been widely investigated in the academic literature, but the issue of the incorporation of uncertainty has received a growing research attention only in the last 15 years. Two alternative approaches have been proposed to handle the problem.

In the first one, the duration of each activity is assumed to be a random variable which follows known probability distribution functions and the scheduling problem is viewed as a multistage decision process, where decisions are made each time new information becomes available.

Scheduling is done by policies that define, at decision point, appropriate actions concerning the choice of activities that should be executed next and the objective is typically the minimization of the expected makespan. First introduced in [56], priority policies schedule activities according to a given priority order. While easy to define and implement, they have been abandoned since the so-called Graham anomalies may occur [32].

Preselective policies were then introduced by Igelmund and Radermacher [38]. Germane to these policies is the concept of minimal forbidden set defined as the minimum cardinality set of activities, without precedence constraints, whose total resource consumption exceeds the resource availability. A preselective policy defines for each minimal forbidden set a (preselected) activity to be postponed in order to solve potential resource conflicts. The partial order of precedence constraints induces a digraph which has a node for each activity and for each waiting condition related to preselected activities.

A combination of priority and preselective policies has been proposed by Möhring and Stork [51], who studied the so-called linear preselective policies where an activity is selected to be delayed and the choice respects the order imposed by a priority list. Since each linear preselective policy is also a preselective policy, linear preselective policies inherit the analytic properties of being monotone and continuous. These properties were further exploited in [59] to develop a branch-and-bound procedure equipped with dominance rules and different branching schemes to efficiently compute an optimal preselective policy.

More recently, a new class of policies, called preprocessor policies, have been proposed in [5]. A priori sequencing decisions resolve some, but not necessarily all, resource conflicts in a preprocessing phase, while the remaining conflicts are dynamically resolved during project execution. In particular, a preprocessor policy is defined by a set of activity pairs (which adds extra precedence relations between activities) and an ordered list used by a priority based policy to solve conflicts during project execution.

Policies have also been used for determining predictive activity starting times, with the objective of minimizing costs related to positive and negative deviations of actual starting times, from the predicted ones, and to penalties/bonuses associated with late/early project completion. Deblaere et al. [24] proposed a methodology for the determination of a project execution policy and a vector of predictive activity starting times. Both expected activity starting time deviations and penalties or bonuses associated with late or early project completion are minimized in objective function. For solving the problem, the authors proposed a combination of four descent procedures that heavily rely on simulation for the evaluation of the objective function.

Since the stochastic RCPSP is challenging from both a theoretical and computational point of view, in the literature it has often been solved by means of tailored heuristic approaches. Golenko and Gonik [30] presented a heuristic algorithm where resource conflicts are resolved by a zero–one integer programming problem. The problem aims at maximizing the total contribution of the accepted activities to the expected project duration, where such contribution is defined as the product of the average duration of the activities and their probability of being on the critical path, calculated via simulation.

Starting from the concept of critical chain introduced by Goldratt [29], Rabbani et al. [54] presented a new heuristic implementing backward pass scheduling for feeding-in resources, with the objective of minimizing the expected project duration and its variance. Similarly to the work of [30], the solution of a zero–one integer programming approach is suggested to allocate the resources, considering that the activities with the greatest probability to be on the critical chain and the greatest correlation with the project variance are fed-in first. Tsai and Gemmil [61] proposed a tabu search based heuristic, which uses multiple tabu lists, randomized short-term memory, and multi-start diversification mechanism. Later on, Ballestín [7] developed regret-based biased random sampling procedures and embedded them into a genetic algorithm. A GRASP-heuristic able to produce high-quality solutions for multiple possible objective functions is proposed [8]. Bruni et al. [19] presented a chance-constrained based heuristic to build baseline schedules with minimum makespan able to absorb dynamic variations of activity durations.

Within the stochastic programming context, a two-stage integer linear stochastic model has been proposed in [66]. Target times are determined in the first stage followed by the development of a detailed project schedule in the second stage. The two-stage stochastic model aims at minimizing the cost of project completion and expected penalty associated with starting time deviations