



## Discrete Optimization

## Bi-Objective Multi-Mode Project Scheduling Under Risk Aversion



Walter J. Gutjahr\*

Department of Statistics and Operations Research, University of Vienna, Oskar-Morgenstern-Platz 1, A-1090 Wien, Austria

## ARTICLE INFO

## Article history:

Received 5 June 2014

Accepted 3 May 2015

Available online 9 May 2015

## Keywords:

Project scheduling

Multi-objective optimization

Stochastic optimization

Risk aversion

Stochastic dominance

## ABSTRACT

The paper proposes a model for stochastic multi-mode resource-constrained project scheduling under risk aversion with the two objectives makespan and cost. Activity durations and costs are assumed as uncertain and modeled as random variables. For the scheduling part of the decision problem, the class of early-start policies is considered. In addition to the schedule, the assignment of execution modes to activities has to be selected. To take risk aversion into account, the approach of optimization under multivariate stochastic dominance constraints, recently developed in other fields, is adopted. For the resulting bi-objective stochastic integer programming problem, the Pareto frontier is determined by means of an exact solution method, incorporating a branch-and-bound technique based on the forbidden set branching scheme from stochastic project scheduling. Randomly generated test instances, partially derived from a test case from the PSPLIB, are used to show the computational feasibility of the approach.

© 2015 Elsevier B.V. and Association of European Operational Research Societies (EURO) within the International Federation of Operational Research Societies (IFORS). All rights reserved.

## 1. Introduction

The Resource-Constrained Project Scheduling Problem (RCPSp) has turned out as a useful and flexible modeling approach in several branches of project management (see, e.g., Brucker, Drexler, Möhring, Neumann, and Pesch, 1999). For situations where projects have to be carried out under limited resources, the RCPSp model allows the determination of optimal schedules with respect to an objective such as the makespan, i.e., the total project completion time. Despite the RCPSp's large versatility, however, there are aspects of real-life project scheduling that are not covered by the classical formulation of this problem, which has led to different extended versions. Three of the most important aspects of this kind are: (i) the necessity to take uncertain activity durations into account, (ii) the case where for single activities of a project, there is a choice between different modes, and (iii) the occurrence of further objectives in addition to the makespan, for example, cost (or profit) not considered as a constraint, but as another objective.

A large number of articles (cf. the surveys Hartmann and Briskorn, 2010; Herroelen and Leus, 2005) have dealt with these three complicating aspects in separation from each other, which has led to (i) stochastic, (ii) multi-mode or (iii) multi-objective generalizations of the RCPSp, respectively. However, a work addressing all three aspects in a combined model seems to be missing. To develop such a model and a suitable solution technique is the first aim of the present paper.

The second aim concerns the aspect of risk aversion. The most common approaches of stochastic project scheduling suppose a risk-neutral project manager, i.e., a decision maker optimizing expected values of outcomes. However, in the practice of project management, this is often not sufficient: special care needs to be taken to avoid extreme values of time and cost, even if their occurrence has a comparably small probability. To represent risk aversion in our optimization model, we shall resort to an approach recently developed in other fields, namely stochastic optimization under multivariate dominance constraints (see Dentcheva & Ruszczyński, 2009). This method fixes a reference solution (e.g., a state-of-the-art solution) and solves then an optimization problem on the constraint that the solution to be proposed stochastically dominates the reference solution. Contrary to the standard applications of this method, the optimization problem under consideration will be *bi-objective* in our case. Its entire efficient frontier – including so-called non-supported solutions – will be determined.

In total, we shall formulate a stochastic multi-mode RCPSp under risk aversion with the two objectives makespan and cost, and compute its Pareto frontier by means of an exact optimization method. The proposed method will make use of (i) an algorithm for optimization under multivariate dominance constraints, developed by Homem-de Mello and Mehrotra (2009), and of (ii) a problem-specific branch-and-bound technique for stochastic project scheduling investigated by Stork (2001). It will be shown that by a suitable modification and integration of these algorithms and by putting them into the context of the well-known epsilon-constraint method for multi-objective optimization, instances of nontrivial size of the considered complex problem can still be solved to optimality.

\* Tel.: +43 14277 38632.

E-mail address: [walter.gutjahr@univie.ac.at](mailto:walter.gutjahr@univie.ac.at)

The paper is organized as follows. [Section 2](#) gives a survey on related work and on the features distinguishing our approach from the available literature. [Section 3](#) formulates the investigated bi-objective stochastic optimization model in mathematical terms, and [Section 4](#) presents the suggested solution algorithms. In [Section 5](#), computational results are presented. [Section 6](#) contains concluding remarks.

## 2. Related literature

Methods of *stochastic project scheduling* have been intensely investigated in the literature, starting with works by Radermacher, Igelmund, Möhring, Stork and others (see, e.g., [Igelmund and Radermacher, 1983](#); [Möhring and Stork, 2000](#); [Stork, 2001](#)). Contrary to deterministic project scheduling problems, where a solution is described by a vector of starting times of activities, a solution to a stochastic project scheduling problem is a (dynamic) *policy*. It turns out that optimization on the class of *all* dynamic scheduling policies for a given stochastic scheduling problem is usually not a viable approach, partly because of the computational complexity of finding the optimal policy, partly because of the difficulties related to the description and practical implementation of a general dynamic policy. Therefore, different subclasses of policies have been proposed. Among them, let us mention the class of *early-start policies* (ES policies), the class of *activity-based priority policies*, and the class of *resource-based priority policies*. A thorough classification and description of project scheduling policies can be found in [Stork \(2001\)](#). In the present paper, we shall choose the framework of ES policies.

Recent research in stochastic project scheduling focuses on several issues, some of which will be outlined in the following. [Ballestin \(2007\)](#) discusses the circumstances under which it is advisable to use stochastic instead of deterministic scheduling methods and develops algorithmic solution techniques. [Zhu, Bard, and Yu \(2007\)](#) address the problem of setting target due dates by using a two-stage stochastic programming framework. [Tereso, Araujo, Moutinho, and Elmaghraby \(2008\)](#) investigate dynamic programming as well as diverse metaheuristic algorithms on a special stochastic resource allocation problem in project scheduling.

[Ballestin and Leus \(2009\)](#) present a comprehensive investigation of the RCPSP under stochastic activity durations, considering diverse objective functions such as the expected makespan, the makespan standard deviation and the probability of meeting a due date. As the computational solution technique, a greedy randomized adaptive search procedure (GRASP) is applied. Their study is of special relevance for the topic of the current paper, since they explicitly address the issue of risk aversion. From the numerical results on correlations between the different objective function values, they conclude that it suffices to focus on the expected makespan (which would be a risk-neutral consideration), since the risk-related measures are strongly correlated with the expected makespan anyway. Let us emphasize, however, that the investigations in [Ballestin and Leus \(2009\)](#) refer to the *single-mode* RCPSP, and that “cost” is not taken into account as an *objective*. In the present paper, we will turn to the multi-mode RCPSP, representing the total cost of the activities in the chosen modes as a separate objective function. This leads to a natural tradeoff not only between makespan and cost, but (as we shall argue in [Section 3.4](#)) also between low-risk, medium-expected-effort solutions on the one hand and high-risk, low-expected-effort solutions on the other hand. In such a context, the decisions preferred by a risk-averse decision maker may be completely different from those that are optimal under a risk-neutral stance.

[Ashtiani, Leus, and Aryanezhad \(2011\)](#) show that by introducing a preprocessing phase where some sequencing decisions are already made *a priori* while the remaining decisions are made dynamically, the performance of solution algorithms for the RCPSP with stochastic activity durations can be considerably improved. [Deblaere, Demeulemeester, and Herroelen \(2011a\)](#) combine the deployment of

a scheduling policy with the determination of a vector of predictive starting times of the single activities. A “policy execution cost” composed of expected penalties for earliness and tardiness of the single activities is minimized.

[Leus and Herroelen \(2004\)](#) and [Leus \(2011\)](#) focus on ES policies and use resource flow variables to represent resource-allocation decisions, which allows the derivation of theoretical results on optimal ES policies. Furthermore, in [Leus \(2011\)](#), an alternative to the “forbidden set branching scheme” from [Stork \(2001\)](#) for enumerating feasible ES policies is proposed by the introduction of a binary branching strategy. [Artigues, Leus, and Nobibon \(2013\)](#) turn from stochastic optimization to robust optimization by minimizing, instead of the expected makespan, the maximum absolute regret over a set of scenarios for the durations of the activities. The two last-mentioned articles are especially interesting in the context of the present paper since we rely here on ES policies as well. Furthermore, let us note that also [Artigues et al. \(2013\)](#) implicitly presupposes a risk-averse attitude of the decision maker, but the model in [Artigues et al. \(2013\)](#) puts less emphasis on the “average situation” than the model presented here where the goal of avoiding too risky decisions is balanced with that of keeping the *expected* makespan low.

In the absence of resource constraints, *multi-mode* versions of the project scheduling problem have been investigated under the terms *time-cost tradeoff problem* or *activity crashing problem*, a problem class for which also stochastic variants exist (see, e.g., [Gutjahr, Strauss, and Wagner, 2000](#)). For the resource-constrained deterministic problem variant, a wealth of literature is available. Most but not all of these articles rely on heuristic solution techniques (see, e.g., [Bouleimen and Lecocq, 2003](#); [Coelho and Vanhoucke, 2011](#); [Deblaere, Demeulemeester, and Herroelen, 2011b](#); [Jozefowska, Mika, Rozycki, Waligora, and Weglarz, 2001](#)). Much less well-investigated is the stochastic multi-mode problem variant: only very few articles dealing with this subject can be found. [Tereso, Araujo, and Elmaghraby \(2004\)](#) and [Tereso et al. \(2008\)](#) treat the resource allocations to the activities as decision variables; different resource allocations incur different costs. The “work contents” of the activities are considered as random variables. [Chen, Zhang, Liu, and Liu \(2010\)](#) consider a stochastic project scheduling problem with uncertain activity durations where each activity  $i$  is associated with a set  $M_i$  of execution modes. A solution is given by a permutation of the activities and a vector of execution modes. A serial schedule generation scheme is used to compute a concrete schedule for the activities, given a solution and a realization of the random variables. As the objective function, the expected net present value of the resulting cash flows is chosen. The model is solved by means of the ant colony optimization metaheuristic. The present paper has some features in common with [Chen et al. \(2010\)](#), but differs from [Chen et al. \(2010\)](#) by (i) considering makespan and cost as two separate objective functions whose Pareto frontier is to be determined, by (ii) replacing the (risk-neutral) expected-value consideration by a risk-averse consideration, and (iii) by solving the resulting problem exactly instead of heuristically.

Also [Muller \(2011\)](#) presents a multi-mode stochastic project scheduling problem. In their formulation, however, also *non-renewable* resources are considered, and uncertainty does not reside in the duration of the activities, but instead in the non-renewable resource requirements of each mode. In the model, the stochastic aspect is dealt with by a chance constraint. A branch-and-cut technique is used for the computational solution of the resulting conic quadratic integer program. As in [Chen et al. \(2010\)](#), the model is single-objective.

[Godinho and Branco \(2012\)](#) study a multi-mode stochastic scheduling problem with a weighted sum of expected cost and expected tardiness as the objective function. Compared to [Chen et al. \(2010\)](#), their model does not consider resource constraints. On the other hand, it provides additional flexibility in the considered scheduling policies insofar as the execution modes need not to be

fixed in advance, but are allowed to be determined dynamically. A special (restricted) class of policies is defined to represent the dependence between starting times of activities and chosen modes. A meta-heuristic called “electromagnetism heuristic” is used for the computational solution. As the two former-mentioned articles, the papers stick to a risk-neutral, single-objective consideration.

Several articles on *multi-objective* versions of the RCPSP have been published. We shall not go into detail concerning this strand of the literature, since most of these papers deal with deterministic models and have therefore not too much in common with the present investigation. Some articles include *robustness* issues in the multi-objective model. As an example, let us mention Abbasi, Shadrokh, and Arkat (2006), where the first objective is makespan and the second objective is a floating-time criterion related to schedule reliability. The authors use a weighted-sum approach to deal with the two objective functions. Closer to the present investigation is Kilic, Ulusoy, and Serifoglu (2008) where a bi-objective model with expected makespan and expected total cost as objectives is presented. The authors apply a multi-objective genetic algorithm for the solution of the model. Contrary to the model in the present paper, Kilic et al. (2008) does not consider resource constraints, and it restricts the consideration to expected values, i.e., does not assume a decision maker with a risk-averse preference structure.

### 3. The model

#### 3.1. The basic RCPSP

Let us start by outlining the simple deterministic RCPSP. After that, three (cumulative) extensions leading to the complete model under consideration shall be explained step by step. We adopt the notation in Ballestin and Leus (2009) and Leus (2011). A project consists of a set  $V = \{0, \dots, n+1\}$  of activities, where 0 (project start) and  $n+1$  (project end) are dummy activities. For each activity  $i$ , its duration  $d_i \geq 0$  is given. Dummy activities do not take time, that is,  $d_0 = d_{n+1} = 0$ . The vector  $(d_1, \dots, d_n)$  of durations of non-dummy activities is denoted by  $d$ . Precedence constraints are given by a set  $A \subset V \times V$  of pairs  $(i, j)$  of activities  $i, j \in V$ , where  $(i, j) \in A$  means that activity  $i$  must be completed before activity  $j$  can start. We assume that  $A$  forms a precedence relation, which is a strict partial order on  $V$  (an irreflexive and transitive relation). A representation by an *activity-on-node* graph is used. The graph with node set  $V$  and edge set  $A$  is denoted by  $G(V, A)$  or, whenever we would like to add the information on the durations assigned to the nodes, by  $G(V, A, d)$ .

Moreover, it is supposed that resource constraints with respect to a certain number of renewable resource types are given. The availability of resource type  $k$  is defined by  $a_k \in \mathbb{N}$  ( $k = 1, \dots, K$ ). During its execution, activity  $i$  consumes  $r_{ik}$  units of resource type  $k$  ( $i = 1, \dots, n$ ;  $k = 1, \dots, K$ ). The dummy activities 0 and  $n+1$  do not consume resources.

A *schedule*  $s$  consists of a vector  $(s_0, \dots, s_{n+1})$  of starting times of activities 0,  $\dots$ ,  $n+1$ . Schedule  $s$  is called *time-feasible* if it satisfies the constraints  $s_j + d_j \leq s_i \quad \forall (i, j) \in A$ . For a given schedule  $s$ , let  $\mathcal{A}(s, t) = \{i \in V \mid s_i \leq t < s_i + d_i\}$  denote the set of activities that are in progress at time  $t$  ( $t = 0, 1, \dots$ ). Schedule  $s$  is called *resource-feasible* if  $\sum_{i \in \mathcal{A}(s, t)} r_{ik} \leq a_k$  for all  $k$  and  $t$ . A schedule is *feasible* if it is both time-feasible and resource-feasible. The most usual objective function considered for the RCPSP is the *makespan*, given by  $d_{n+1}$  (the time of the event “project end”).

The resource constraints can be represented by means of minimal forbidden sets. For any given precedence relation  $E$ , a set  $U \subseteq V$  of activities is called a *forbidden set* if it is an anti-chain of  $E$  (i.e., if  $(i, j)$  is not contained in  $E$  for any two activities  $i, j \in U$ ), and  $\sum_{i \in U} r_{ik} > a_k$  for at least one  $k \in K$ . Informally speaking, a forbidden set consists of activities that cannot be executed simultaneously, given the resource constraints, although they could be executed simultaneously

with respect to the time constraints. An inclusion-minimal forbidden set is called a *minimal forbidden set* (MFS). The family of MFSs with respect to  $E$  will be denoted by  $\mathcal{F}(E)$ . Using the notion of MFSs, we can express resource-feasibility by saying that a schedule  $s$  is resource-feasible if and only if there is no  $U \in \mathcal{F}(A)$  such that  $U \subseteq \mathcal{A}(s, t)$  for some  $t$ .

The RCPSP will be successively extended in this paper in three respects: First, the deterministic RCPSP is generalized to the *stochastic* RCPSP (S-RCPSP). From the different alternatives studied in the literature, we choose the use of early-start policies for this generalization and focus on the (expected) makespan as an objective function. Second, a *multi-mode* variant of the problem will be considered. This introduces (expected) cost as a second objective function. The two objective functions shall not be aggregated, but considered simultaneously in a bi-objective consideration. Third, instead of a risk-neutral interpretation of the problem, where the expected values of makespan and cost would be the only relevant criteria for the decision, we shall assume a *risk-averse* decision maker. Risk aversion both refers to makespan and cost. Since these two quantities are stochastically correlated, a difficult bi-objective stochastic optimization problem under risk aversion is obtained. Few solution concepts are known for such problems. We shall adopt the recently developed concept of optimization under *multivariate stochastic dominance constraints* (Dentcheva & Ruszczyński, 2009). This approach assumes that to a concrete given problem instance, an acceptable reference solution is known. A constraint is added to the problem formulation requiring that the output solution stochastically dominates the reference solution based on a given multivariate stochastic dominance relation. In this way, too risky solutions are excluded from consideration even if they look attractive from a pure expected-value point of view.

The resulting problem will be called the Risk-Averse Multi-Objective Stochastic Multi-Mode RCPSP or short RAMOS-MRCPPSP. Its basic structure is as follows:

$$\min(\mathbb{E}[\text{makespan}(X)], \mathbb{E}[\text{cost}(X)])$$

s.t.  $X$  stochastically dominates  $X_0$

where  $X_0$  is the reference solution and the bi-objective minimization is interpreted in the sense of the determination of the Pareto-efficient solutions. Dealing with two objectives in such a formulation is a novel feature of this paper compared to previous publication on optimization under multivariate stochastic dominance constraints, which have a single objective function in the upper line. In the following three subsections, the gradual extension of the RCPSP to the RAMOS-MRCPPSP will be specified in precise formal terms.

#### 3.2. Stochastic extension

In the stochastic extension of the basic RCPSP, the deterministic durations  $d_i$  are replaced by random durations  $D_i$  ( $i = 1, \dots, n$ ), which gives a random vector  $D = (D_1, \dots, D_n)$ . The random variables  $D_i$  can be correlated. It will be assumed that the joint distribution of the durations  $D_i$  is a finite discrete distribution represented by a set of  $N$  equally likely<sup>1</sup> scenarios  $\omega_1, \dots, \omega_N$ . Thus, each scenario  $\omega_v$  is supposed to occur with probability  $1/N$ . In scenario  $\omega_v$ , the vector of durations is  $D^v = (d_1^v, \dots, d_n^v)$ . We assume  $d_i^v \in \mathbb{N}_0$  for all  $i$  and  $v$ .

By employing a scenario-based model, we choose a mainstream approach from the stochastic programming literature. One of the advantages of this approach is that it can represent correlations between random variables. Very frequently, activity durations are not independent; in construction applications, e.g., weather conditions

<sup>1</sup> A generalization of the presented approach to the case of scenarios with given probabilities that need not to be equal is possibly not too difficult. Let us mention, however, that the equiprobable case is of prevailing practical relevance insofar as it is the case one faces whenever an underlying probability distribution is approximated by drawing a sample of scenarios according to the given distribution.



or events on the labor market introduce positive correlations. While a simulation model for generating reasonable sample realizations even for correlated durations may be easy to set up, a combination of such a simulation component with an optimization algorithm can be too time-consuming. This makes it advantageous to separate the simulation from the optimization by means of a *a priori* sample generation. For the latter, well-developed methodology exists. Of course, this approach comes at the price that the stochastic model considered during the optimization is only an approximation to the “true” stochastic model.

To describe solutions, we consider the family of *early-start policies* (ES policies).<sup>2</sup> Let  $T(E)$  denote the transitive closure of relation  $E$ . An ES-policy is determined by a set  $X \subseteq (V \times V) \setminus A$  of activity pairs with the properties that  $\mathcal{F}(T(A \cup X)) = \emptyset$  and that the graph  $G(V, A \cup X)$  is acyclic. Such an  $X$  is called a *sufficient selection*. In less formal terms, a sufficient selection is an extension of the precedence constraints achieving the purpose that no resource-infeasible subset of activities can be started simultaneously anymore. In this way, the resource constraints are reduced to time constraints with respect to the extended precedences. We are looking for the “best” extension of the precedences serving this purpose. The ES policy determined by sufficient selection  $X$  is the mapping of each possible duration vector  $d = (d_1, \dots, d_n)$  to a feasible schedule  $s(X, d)$ , where  $s(X, d)$  is given by the starting times  $s_i(X, d) = \text{length of the longest path from } 0 \text{ to } i \text{ in } G(V, A \cup X, d)$  (cf. Artigues et al., 2013 for details).

In the stochastic case, the makespan  $D_{n+1}$  under the chosen ES policy becomes a random variable. A well-investigated version of the stochastic RCPSP consists in minimizing the *expected makespan*  $E[D_{n+1}]$ .

### 3.3. Multi-mode extension

In addition to the previous assumptions, we shall assume from now on that for each activity  $i$ , the project manager can choose an *execution mode* (short: mode) from a finite set  $M_i$ . The mode defines the specific way how activity  $i$  is performed. In particular, the following assumptions are made:

- The durations of the activities depend on the mode, i.e., we have  $D_i = D_{im}$  ( $m \in M_i$ ). The realizations of the random variable  $D_{im}$  in scenarios  $\omega_1, \dots, \omega_N$  are denoted by  $d_{im}^1, \dots, d_{im}^N$ , respectively.
- To each mode  $m \in M_i$ , an (execution) cost  $C_{im}$  is assigned. Also the cost  $C_{im}$  is a random variable. Its realizations in scenarios  $\omega_1, \dots, \omega_N$  are denoted by  $c_{im}^1, \dots, c_{im}^N$ , respectively.
- The resource consumptions  $r_{ik}$  can depend on the mode:  $r_{ik} = r_{ik}(m)$  for  $m \in M_i$ .

A solution  $X = (p, \mathbf{m})$  of the stochastic multi-mode RCPSP obtained in this way contains two components: (1) an ES-policy given by the sufficient selection  $p$ , and (2) a vector  $\mathbf{m} = (m_1, \dots, m_n)$  assigning to each activity  $i$  a mode  $m_i \in M_i$  ( $i = 1, \dots, n$ ).

Let  $\mathcal{M} = M_1 \times \dots \times M_n$  denote the set of all mode combinations  $\mathbf{m}$ . Set  $r_{ik}(\mathbf{m}) = r_{ik}(m_i)$  ( $i = 1, \dots, n; k = 1, \dots, K$ ). By  $\Pi_{\mathbf{m}}$ , we denote the set of sufficient selections for the given graph  $G(V, A)$  and the given mapping  $E \mapsto \mathcal{F}(E)$  of precedence relations to MFSs under mode combination  $\mathbf{m}$ . Note that since the resource consumptions depend on  $\mathbf{m}$ , also the set of MFSs for a given precedence relation  $E$  can

depend on  $\mathbf{m}$ , and hence also the set  $\Pi_{\mathbf{m}}$ . Then the search space of the extended problem is  $\mathcal{X} = \{(p, \mathbf{m}) \mid p \in \Pi_{\mathbf{m}}, \mathbf{m} \in \mathcal{M}\}$ .

For the sake of consistency with the usual framework of expected utility theory, we formulate the *objective functions* as functions to be maximized, setting

$$\begin{aligned} f_1^v(p, \mathbf{m}) &= -(\text{makespan of solution } (p, \mathbf{m}) \text{ in scenario } v), \\ f_2^v(p, \mathbf{m}) &= -(\text{cost of solution } (p, \mathbf{m}) \text{ in scenario } v). \end{aligned}$$

The makespan  $g^v(p, \mathbf{m})$  of solution  $(p, \mathbf{m})$  in scenario  $v$  can be obtained as the solution value of the critical path problem, and from this value, the first objective can be immediately computed through  $f_1^v(p, \mathbf{m}) = -g^v(p, \mathbf{m})$ .

Concerning the second objective, observe that  $f_2^v(p, \mathbf{m})$  only depends on  $\mathbf{m}$ :

$$f_2^v(p, \mathbf{m}) = f_2^v(\mathbf{m}) = - \sum_{i=1}^n \sum_{j=1}^{m_i} c_{ij}^v m_{ij} \quad (1)$$

with  $m_{ij} = 1$  if  $m_i = j$  and  $m_{ij} = 0$  otherwise ( $j = 1, \dots, |M_i|$ ;  $i = 1, \dots, n$ ). We have the obvious constraints  $\sum_{j=1}^{|M_i|} m_{ij} = 1$  ( $i = 1, \dots, n$ ). By collecting all costs in a column vector,

$$\mathbf{c}^v = (c_{11}^v, \dots, c_{1, |M_1|}^v, \dots, c_{n1}^v, \dots, c_{n, |M_n|}^v)',$$

and doing the same for the decision variables  $m_{ij}$ , we can rewrite (1) as the inner product<sup>3</sup>

$$f_2^v = -(\mathbf{c}^v)' \mathbf{m}. \quad (2)$$

In total, the following bi-objective stochastic optimization problem is obtained:

$$\begin{aligned} \max \quad & (F_1(p, \mathbf{m}), F_2(\mathbf{m})) \\ \text{s.t.} \quad & p \in \Pi_{\mathbf{m}}, \quad \mathbf{m} \in \mathcal{M} \end{aligned} \quad (3)$$

where  $F_1(p, \mathbf{m})$  and  $F_2(\mathbf{m})$  denote the random variables with values  $f_1^v(p, \mathbf{m})$  and  $f_2^v(\mathbf{m})$ , respectively, with  $v$  being drawn uniformly at random from  $\{1, \dots, N\}$ .

Obviously, (3) is still under-specified as a (bi-objective) mathematical program, since it is not clear (i) in which sense the *random* functions should be maximized, and (ii) how to deal with the presence of two objective functions. The simplest way to address (i) would be to take *expectations*, i.e., to replace the objective function line of (3) by  $\max(E[F_1(p, \mathbf{m})], E[F_2(\mathbf{m})])$ . For this variant, the set of *Pareto-efficient* solutions could be determined, which would then answer also question (ii) above. However, relying only on expectations of benefits would be a *risk-neutral* standpoint which we do not adopt in this paper.

### 3.4. Risk-averse extension

Whereas in single-mode stochastic project scheduling, minimization of the expected makespan usually also keeps the variance of the makespan and hence the risk of very high makespan realizations low, which makes a risk-neutral consideration sufficient (cf. Ballestin and Leus, 2009), in multi-mode stochastic project scheduling, tradeoffs between expected values and risks are notorious to occur in practice. For example, many activities can be carried out either (i) in a cheap, but not very reliable mode (e.g. “quick-and-dirty”-programming in software engineering, or building under improper time pressure in construction), or alternatively (ii) in a sound, but more expensive mode. In case of option (i), both expected cost and

<sup>2</sup> A main motivation for this choice is that ES policies allow elegant mixed-integer programming formulations based on resource flows (Leus, 2011; Leus & Herroelen, 2004). We do not require this type of formulations in the present approach, but they could possibly be helpful in future attempts to improve the optimization algorithms. From the viewpoint of the achieved expected makespan, ES policies seem to be at least competitive (if not better) compared to usual alternatives as activity-based or resource-based policies (see Stork, 2001). Let us mention, however, that by the more complex preprocessor policies (PP policies), quality improvements over ES policies have been obtained (Ashtiani et al., 2011).

<sup>3</sup> Abusing notation, we interpret  $\mathbf{m}$  alternatively as the vector  $(m_1, \dots, m_n)$  of integer mode indices or as the vector of binary mode indicator variables  $(m_{ij})$ . Since the information content of both representations is the same, this does not cause any ambiguity. In particular,  $\mathcal{M}$  can be considered as the set of binary variables  $(m_{ij})$  satisfying  $\sum_j m_{ij} = 1$  for all  $i$ .

expected makespan are typically low, but as a consequence of possibly necessary re-work and laborious ex-post corrections, the variances of cost and makespan can be high. Contrary, in case of option (ii), both expected cost and expected makespan will be comparably higher; however, the variances of cost and makespan can be kept low. In such a situation, a risk-neutral decision maker (looking only at expected values) will prefer option (i), whereas a risk-averse decision maker will prefer option (ii).

For dealing with multi-objective stochastic optimization problems under risk aversion, the use of *multivariate stochastic dominance constraints* has recently been proposed by Dentcheva and Ruszczyński (2009). In the special case of two objective functions, this approach works as follows: A *bivariate stochastic order*  $\succeq$  on the set of two-dimensional random vectors and a deterministic objective function  $h(x)$  are fixed. Moreover, for the given problem instance, a *reference solution* with random (bivariate) outcome  $\mathbf{Y}$  is chosen. The reference solution should be conservative with respect to risk. One tries now to improve the reference solution without sacrificing its property of being not too risky. In formal terms, as a concretization of the (under-specified) problem  $\max\{(f_1(x, \omega), f_2(x, \omega)) \mid x \in \mathcal{X}\}$ , the problem

$$\begin{aligned} \max \quad & h(x) \\ \text{s.t.} \quad & (f_1(x, \omega), f_2(x, \omega)) \succeq \mathbf{Y}, \quad x \in \mathcal{X} \end{aligned} \quad (4)$$

is introduced. In this paper, we extend this approach by modifying (4) to a bi-objective problem in the sense of the Pareto-efficiency concept as follows:

$$\begin{aligned} \max \quad & (\mathbf{E}[f_1(x, \omega)], \mathbf{E}[f_2(x, \omega)]) \\ \text{s.t.} \quad & (f_1(x, \omega), f_2(x, \omega)) \succeq \mathbf{Y}, \quad x \in \mathcal{X} \end{aligned} \quad (5)$$

In other words, instead of the auxiliary objective function  $h(x)$ , we insert the two expectations of  $f_1$  and  $f_2$ , and look for the resulting Pareto frontier. In this way, we can allow ourselves to be risk-neutral in the objective function line because too risky solutions are already filtered out by the stochastic dominance constraint.

In the literature on optimization under multivariate stochastic dominance constraints, a standard choice for the multivariate stochastic order is the *increasing positive linear concave order* (cf. Dentcheva and Ruszczyński, 2009; Gutjahr and Pichler, 2013; Müller and Stoyan, 2002). For random variables  $\mathbf{X} = (X_1, \dots, X_q)$  and  $\mathbf{Y} = (Y_1, \dots, Y_q)$ , this order is defined by

$$\mathbf{X} \succeq_{\mathcal{P}}^{\text{lin}} \mathbf{Y} \Leftrightarrow \mathbf{v}'\mathbf{X} \succeq_{(2)} \mathbf{v}'\mathbf{Y} \quad \forall \mathbf{v} \in \mathcal{P} \quad (6)$$

where  $\succeq_{(2)}$  denotes second-order stochastic dominance<sup>4</sup> and for the domain  $\mathcal{P}$  of weight vectors  $\mathbf{v}$ , the default choice is  $\mathcal{P} = \mathbb{R}_+^q$ . Homem-de Mello and Mehrotra (2009) generalize this definition by allowing  $\mathcal{P}$  to be an arbitrary polyhedral set  $\mathcal{P} \subseteq \mathbb{R}_+^q$ . We adopt this latter, more flexible dominance relation in the present work. More specifically, in our context where  $q = 2$ , we set  $\mathcal{P} = \{(1, w)' \mid a \leq w \leq b\}$  with constants  $a$  and  $b$  ( $0 < a \leq b$ ): Observe that except in the degenerate case where one of the objectives gets weight zero, one may assign weight 1 to the first objective without loss of generality, and that a polyhedral set on the real line is an interval. In this way, we arrive at the problem

$$\begin{aligned} \max \quad & (\mathbf{E}[F_1(p, \mathbf{m})], \mathbf{E}[F_2(\mathbf{m})]) \\ \text{s.t.} \quad & (F_1(p, \mathbf{m}), F_2(\mathbf{m})) \succeq_{\mathcal{P}}^{\text{lin}} \mathbf{Y} \\ & p \in \Pi_{\mathbf{m}}, \quad \mathbf{m} \in \mathcal{M} \end{aligned} \quad (7)$$

where  $\mathbf{Y} = (F_1(p_0, \mathbf{m}_0), F_2(\mathbf{m}_0))$  is the vector composed of the objective function values of the reference solution  $(p_0, \mathbf{m}_0)$ .

**Example 1.** To illustrate the used stochastic dominance concept, let us give the following simple example. Assume that two solutions  $(p_0,$

$\mathbf{m}_0)$  and  $(p_1, \mathbf{m}_1)$  are compared under  $N = 4$  equiprobable scenarios and produce in scenarios 1–4 the values below for makespan and cost. Moreover, assume that the weight  $w$  for cost is in the interval  $[a, b] = [0.1, 1]$ .

	makespan	cost
$(p_0, \mathbf{m}_0)$ :	2, 10, 2, 4	3, 2, 8, 1
$(p_1, \mathbf{m}_1)$ :	4, 5, 5, 5	1, 2, 3, 4

For the purpose of this example, we apply the following criterion (see Roman, Darby-Dowman, & Mitra, 2006) for second-order stochastic dominance in the case of equiprobable scenarios: Sort profits (i.e., negative costs) in nondecreasing order and cumulate the resulting values successively from left to right. Then dominance w.r.t.  $\succeq_{(2)}$  holds between the original vectors iff dominance w.r.t. the ordinary partial order on  $\mathbb{R}^N$  holds between the resulting vectors. (In less formal terms, this means that the first solution dominates the second one w.r.t.  $\succeq_{(2)}$  if it is at least as good in the respective worst case, at least as good in the average of the two worst cases, in the average of the three worst cases, etc., and finally also in the average of all cases.) Setting  $w = 1$  (i.e., giving makespan and cost the same weight 1), we obtain:

	weighted sums	sorted and cumulated negatives
$(p_0, \mathbf{m}_0)$ :	5, 12, 10, 5	−12, −22, −27, −32
$(p_1, \mathbf{m}_1)$ :	5, 7, 8, 9	−9, −17, −24, −29

We see that  $(p_1, \mathbf{m}_1) \succeq_{(2)} (p_0, \mathbf{m}_0)$  under  $w = 1$ . Now consider  $w = 0.1$ , i.e., the case where cost is only given 10 percent of the weight assigned to makespan. Then we get

	weighted sums	sorted and cumulated negatives
$(p_0, \mathbf{m}_0)$ :	2.3, 10.2, 2.8, 4.1	−10.2, −14.3, −17.1, −19.4
$(p_1, \mathbf{m}_1)$ :	4.1, 5.2, 5.3, 5.4	−5.4, −10.7, −15.9, −20.0

Now the two solutions are mutually incomparable: a risk-neutral decision maker would choose the first one, which has the better expected value, while a risk-averse decision maker might choose the second one, which is better in the worst case. In total, for  $[a, b] = [0.1, 1]$ , the second solution does *not* dominate the first solution w.r.t.  $\succeq_{\mathcal{P}}^{\text{lin}}$ . So if  $(p_0, \mathbf{m}_0)$  is the reference solution,  $(p_1, \mathbf{m}_1)$  is not feasible for (7). Note that even if the second solution would dominate the first one both for  $w = a$  and  $w = b$ , it would not yet be guaranteed that dominance also holds in the sense of  $\succeq_{\mathcal{P}}^{\text{lin}}$ , that is, for all weights  $w$  with  $a \leq w \leq b$ . In the next section it will be shown how testing for  $\succeq_{\mathcal{P}}^{\text{lin}}$  is possible without trying infinitely many values of  $w$ .

Problem (7) is solved in the sense of the determination of its *Pareto frontier*. This notion is defined in the usual way: For two points  $\mathbf{z} = (z_1, z_2) \in \mathbb{R}^2$  and  $\mathbf{w} = (w_1, w_2) \in \mathbb{R}^2$ , let us write  $\mathbf{w} \succ \mathbf{z}$  if  $w_i \geq z_i$  with at least one inequality being strict. (Note the difference between the deterministic dominance relation  $\succ$  and the stochastic dominance relation  $\succeq_{\mathcal{P}}^{\text{lin}}$ !) For two objective functions  $f_1$  and  $f_2$  to be maximized, element  $x$  of the feasible set  $\mathcal{X}$  is called *Pareto-efficient* (short: *efficient*) if there is no  $y \in \mathcal{X}$  with  $(f_1(y), f_2(y)) \succ (f_1(x), f_2(x))$ . The set  $S$  of all efficient solutions is called the *Pareto set*, and the image  $f(S) \subseteq \mathbb{R}^2$  of this set in the objective space  $\mathbb{R}^2$  is called the *Pareto frontier*.

**Remark 1.** The proposed method requires the definition of a reference solution. There are several ways to find such a solution. One way would be to generate a set of promising solution candidates by some heuristic technique, to assess them by simulation and/or discussion among experts, and to select the most attractive among them. Another alternative would be to apply a (max–min or min–max-regret) robust optimization approach. A third alternative applies to the situation of a project with fixed structure that is carried out repeatedly under different circumstances; in this case, the reference solution may simply be based on a field-tested standard project plan. Of course, the last-mentioned alternative requires that the project manager has already applied the technology of planning by ES policies in the past, such that the standard plan is available in terms of a formal ES policy.

<sup>4</sup> A possible way to define (univariate) second-order stochastic dominance is by the definition  $X \succeq_{(2)} Y$  iff  $\mathbf{E}[u(X)] \geq \mathbf{E}[u(Y)]$  for each nondecreasing, concave utility function  $u$  for which the expectations exist.

It can happen that the reference solution turns out as “too good” in the sense that as after the restriction to the solutions dominating it, only few solutions (or even only the reference solution itself) remain, while the project manager would prefer to choose from a larger set of solution candidates. An option for dealing with this situation will be presented in Section 5.

#### 4. Computational solution

We shall provide an *exact* solution algorithm for problem (7), but in view of the enormous complexity of the problem, it is clear that some restrictive assumptions on the size of the problem instance have to be made. In particular, we shall assume that the set  $\mathcal{M}$  of possible mode combinations is not yet too large to be completely enumerated. This assumption is frequently realistic in practice: Even if the number  $n$  of activities is large, it is often the case that only a small subset of them can be executed in different modes, which means that  $|\mathcal{M}_i| = 1$  for most activities.

Whereas we assume that complete enumeration of the mode combinations  $\mathbf{m}$  is still feasible, we cannot assume the same for the sufficient selections  $p$ . Their number is growing with an order of  $O(2^{n^2})$  in the number  $n$  of activities, i.e., very fast. Thus, an efficient integer-programming-based algorithm will be required as part of the solution procedure concerning this component of the solution  $\mathcal{X} = (p, \mathbf{m})$ .

To deal with the multi-objective aspect of the problem, we shall apply a simple algorithmic framework based on the *epsilon-constraint method*. (For variants of the epsilon-constraint method, the reader is referred to Laumanns, Thiele, and Zitzler, 2006.) Note that  $E[F_2(\mathbf{m})]$  only depends on the mode combination  $\mathbf{m} \in \mathcal{M}$ . The proposed framework procedure is described in Algorithm 1 below.

**Algorithm 1.** (Framework Procedure).

1. Set  $S = \emptyset$ .
2. Go through all possible levels  $h$  of  $E[F_2(\mathbf{m})]$  in descending order (i.e., ascending order of expected costs  $-E[F_2(\mathbf{m})]$ ). For each  $h$ :
  - (a) Solve
 
$$\max_{p, \mathbf{m}} E[F_1(p, \mathbf{m})] \quad (8)$$

$$\text{s.t. } E[F_2(\mathbf{m})] = h \quad (9)$$

$$(F_1(p, \mathbf{m}), F_2(\mathbf{m})) \succeq_p^{\text{lin}} \mathbf{Y} \quad (10)$$

$$p \in \Pi_{\mathbf{m}}, \mathbf{m} \in \mathcal{M} \quad (11)$$
  - (b) If a solution  $(p, \mathbf{m})$  to (8)–(11) exists and if there is no  $(p', \mathbf{m}') \in S$  with
 
$$(E[F_1(p', \mathbf{m}')], E[F_2(\mathbf{m}')]) \succ (E[F_1(p, \mathbf{m})], E[F_2(\mathbf{m})]),$$
 add  $(p, \mathbf{m})$  to  $S$ .

In Algorithm 1, by the formulas in Section 3.3,

$$E[F_1(p, \mathbf{m})] = -\frac{1}{N} \sum_{v=1}^N g^v(p, \mathbf{m}), \quad E[F_2(\mathbf{m})] = -\frac{1}{N} \sum_{v=1}^N (\mathbf{c}^v)' \mathbf{m}. \quad (12)$$

The loop in the procedure above can be organized as follows: First, sort all mode combinations  $\mathbf{m}$  w.r.t. their expected costs. With each possible cost level  $-h$ , one or several mode combinations  $\mathbf{m}$  are associated. For each of these  $\mathbf{m}$ , find the sufficient selection  $p$  maximizing  $E[F_1(p, \mathbf{m})]$  subject to the constraints (10)–(11). The mode combination  $\mathbf{m}$  giving the best objective function value, together with the corresponding  $p$ , specifies the solution  $(p, \mathbf{m})$  for the current level  $h$ . Evidently, the overall procedure generates the entire Pareto frontier.

For fixed  $\mathbf{m}$ , this gives the following subproblem:

$$\min_p \sum_{v=1}^N g^v(p, \mathbf{m}) \quad (13)$$

$$\text{s.t. } (-[g^v(p, \mathbf{m})], -(\mathbf{c}^v)' \mathbf{m}) \succeq_p^{\text{lin}} (-[g^v(p_0, \mathbf{m}_0)], -(\mathbf{c}^v)' \mathbf{m}_0) \quad (14)$$

$$p \in \Pi_{\mathbf{m}} \quad (15)$$

where  $[\xi^v]$  denotes a random variable with equiprobable realizations  $\xi^1, \dots, \xi^N$ .

For an efficient determination of the Pareto frontier, it should be observed that many (if not most) mode combinations  $\mathbf{m}$  will not produce a feasible solution to (13)–(15). In particular, mode combinations with too high expected costs will not be able to dominate the reference solution, but also low-cost modes may fail to give a solution satisfying (14) in view of possibly large makespan values. Thus, when tackling problem (13)–(15), it is advisable first to test whether a feasible solution exists at all (as we shall see, this decision problem is comparably easier), and to invest the runtime for determining the optimal solution only in the case that feasibility has already been checked. We shall describe these two steps in the following two subsections.

##### 4.1. Feasibility test

For judging whether (14) can be satisfied by a suitable sufficient selection  $p$ , we use the criterion below for second-order stochastic dominance given in Homem-de Mello and Mehrotra (2009): Let  $\xi$  and  $\zeta$  be (univariate) random variables with equiprobable realizations  $\xi^v$  and  $\zeta^v$ , respectively ( $v = 1, \dots, N$ ). Then the statement  $\xi \succeq_{(2)} \zeta$  is equivalent to

$$\sum_{\mu=1}^N (\zeta^v - \xi^\mu)_+ \leq \sum_{\mu=1}^N (\zeta^v - \zeta^\mu)_+ \quad (v = 1, \dots, N).$$

Therefore, (6) can be written in the discrete finite equiprobable case as

$$\begin{aligned} \mathbf{X} \succeq_p^{\text{lin}} \mathbf{Y} &\Leftrightarrow \sum_{\mu=1}^N (\mathbf{v}' \mathbf{Y}^\mu - \mathbf{v}' \mathbf{X}^\mu)_+ \\ &\leq \sum_{\mu=1}^N (\mathbf{v}' \mathbf{Y}^\mu - \mathbf{v}' \mathbf{Y}^\mu)_+ \quad (v = 1, \dots, N) \quad \forall \mathbf{v} \in \mathcal{P}, \end{aligned}$$

where  $\mathbf{X}^v$  and  $\mathbf{Y}^v$  is the  $v$ -th realization of  $\mathbf{X}$  and  $\mathbf{Y}$ , respectively. In our case, the stochastic dominance relation (14) refers to the random vectors  $\mathbf{X} = N \cdot (F_1(p, \mathbf{m}), F_2(\mathbf{m}))$  and  $\mathbf{Y} = N \cdot (F_1(p_0, \mathbf{m}_0), F_2(\mathbf{m}_0))$  with realizations  $\mathbf{X}^v = (-g^v(p, \mathbf{m}), -(\mathbf{c}^v)' \mathbf{m})$  and

$$\mathbf{Y}^v = (-g^v(p_0, \mathbf{m}_0), -(\mathbf{c}^v)' \mathbf{m}_0) =: (\alpha^v, \beta^v),$$

where the numbers  $\alpha^v$  and  $\beta^v$  are constants, since the reference solution is given. Thus, in view of  $\mathbf{v} = (1, w)$ , problem (13)–(15) is feasible if and only if

$$\begin{aligned} \exists p \in \Pi_{\mathbf{m}} : \sum_{\mu=1}^N (\alpha^v + w\beta^v + g^\mu(p, \mathbf{m}) + w(\mathbf{c}^\mu)' \mathbf{m})_+ \\ - \sum_{\mu=1}^N (\alpha^v + w\beta^v - \alpha^\mu - w\beta^\mu)_+ \leq 0 \\ \forall v = 1, \dots, N, \quad w \in [a, b]. \end{aligned} \quad (16)$$

For deciding whether (16) holds, we use Algorithm 1 from Homem-de Mello and Mehrotra (2009) in a modified form. Actually, this algorithm is already designed for optimization problems (as opposed to decision problems), so it would be tempting to apply it directly to (8)–(11). However, since our constraints contain



the makespan values  $g^\mu(p, \mathbf{m})$ , this would require the solution of an optimization problem within the (non-linear) constraints of another optimization problem, which is difficult. For this reason, we choose a slightly different path by first applying Homem-de-Mello's and Mehrotra's algorithm only for deciding on feasibility, using an auxiliary objective function, and introducing the true objective function (13) afterwards (details will be described in the next subsection). The modification of the algorithm required in our context is given below as Algorithm 2.

**Algorithm 2.** (Feasibility Test).

1. Set  $s = 0$ ,  $w = a$ ,  $v = 1$ , and  $W^0 = \{(w, v)\}$ .
2. Solve the problem

$$\min_p \max_{(w,v) \in W^s} \left[ \sum_{\mu=1}^N (\alpha^v + w\beta^v + g^\mu(p, \mathbf{m}) + w(\mathbf{c}^\mu)' \mathbf{m})_+ - \sum_{\mu=1}^N (\alpha^v + w\beta^v - \alpha^\mu - w\beta^\mu)_+ \right] \quad (17)$$

$$\text{s.t. } p \in \Pi_{\mathbf{m}} \quad (18)$$

If the solution value is  $> 0$ , **stop** with the answer “false” to (16). Otherwise, let  $\hat{p}$  be the optimal sufficient selection.

3. Let  $\hat{\alpha}^\mu = -g^\mu(\hat{p}, \mathbf{m})$  and  $\hat{\beta}^\mu = -(\mathbf{c}^\mu)' \mathbf{m}$ . For  $v = 1, 2, \dots, N$ , solve

$$\min_{w,y} \psi_v(w, y) := \sum_{\mu=1}^N y_\mu - \sum_{\mu=1}^N (\alpha^v + w\beta^v - \hat{\alpha}^\mu - w\hat{\beta}^\mu)_+ \quad (19)$$

$$\text{s.t. } a \leq w \leq b \quad (20)$$

$$y_\mu \geq \alpha^v + w\beta^v - \alpha^\mu - w\beta^\mu \quad (\mu = 1, \dots, N) \quad (21)$$

$$y_\mu \geq 0 \quad (\mu = 1, \dots, N) \quad (22)$$

As soon as for some  $v$  in this loop, a negative solution value is obtained, go to Step 4.

4. If in Step 3, the solution values have been nonnegative for all investigated  $v = 1, \dots, N$ , **stop** with the answer “true” to (16). Otherwise, let  $v$  be the index for which a negative solution value has been obtained, and let  $(w, y)$  be the corresponding solution. Add  $w = w^s$  and  $v = v^s$  to  $W^s$  by setting  $W^s = W^s \cup \{(w^s, v^s)\}$ .
5. Set  $s = s + 1$  and go to Step 2.

Let us now comment the single steps of Algorithm 2 in detail:

**Step 1:** The overall algorithm gradually extends a subset  $W^s$  of the set  $[a, b] \times \{1, \dots, N\}$  of all pairs  $(w, v)$  for which the inequality in (16) has to be satisfied. Using  $s = 0, 1, \dots$  as the iteration counter, we initialize  $W^0$  as  $\{(a, 1)\}$ .

**Step 2:** Let  $\text{relax}(W^s)$  denote the relaxation of (16) to the weight-index pairs  $(w, v)$  in  $W^s$ . Then  $\text{relax}(W^s)$  is solvable if and only if (17)–(18) has a solution value  $\leq 0$ . So if the solution value of (17) and (18) is positive,  $\text{relax}(W^s)$  and therefore also (16) is false. Of course the converse does not hold: If the solution value of (17) and (18) turns out as nonpositive, which means that a  $\hat{p} \in \Pi_{\mathbf{m}}$  exists such that the inequality in (16) is satisfied at least for all  $(w, v) \in W^s$ , we have still to check whether there is not any other pair  $(w, v)$  for which it is violated (this is done in Step 3). To solve (17) and (18), we use a branch-and-bound technique which will be presented later.

**Step 3:** This step tries to disprove that the just found  $\hat{p} \in \Pi_{\mathbf{m}}$  satisfies the requirement of (16) by searching a weight  $w$  and a corresponding scenario index  $v$  such that the inequality in (16) is violated. The fact that (19)–(22) can be used for this purpose results as in the proof of Theorem 1 in Homem-de

Mello and Mehrotra (2009): The inequality in (16) is violated for the given  $\hat{p}$  if and only if there exists an index  $v$  such that

$$\min_{w \in [a,b]} \left( \sum_{\mu=1}^N (\alpha^v + w\beta^v - \alpha^\mu - w\beta^\mu)_+ - \sum_{\mu=1}^N (\alpha^v + w\beta^v - \hat{\alpha}^\mu - w\hat{\beta}^\mu)_+ \right) \quad (23)$$

has a negative value. The objective function in (23) is a difference of two convex functions in  $w$ . To obtain a concave minimization problem, one introduces the auxiliary variables  $y_1, \dots, y_N$  defined as  $y_\mu = (\alpha^v + w\beta^v - \alpha^\mu - w\beta^\mu)_+$  ( $\mu = 1, \dots, N$ ). When substituting them in (23), it suffices to impose the constraints (21) and (22) on them, since they are automatically minimized under these constraints. In this way, (23) turns into (19)–(22). Whenever a solution  $(w, y)$  with a negative solution value of (19) is found, we know that the corresponding  $w$  violates the  $v$ -th condition of (16) for the given  $\hat{p}$  and thus disproves  $\hat{p}$  as a solution of (16). If no such  $(w, y)$  is found,  $\hat{p}$  solves (16).

**Step 4:** It is clear that we can stop (with answer “true”) if all solution values of (19) have turned out as nonnegative. Otherwise, we have to cut off  $\hat{p}$  from the set of candidate solutions and to repeat the procedure. Since  $\hat{p}$  violates the constraints of (16) w.r.t. weight  $w$  and index  $v$ , it is sufficient to add the pair  $(w, v)$  to the set  $W^s$  containing the pairs w.r.t. which (17) and (18) is solved.

**Step 5:** This step closes the loop which, by the finiteness of  $\Pi_{\mathbf{m}}$ , is terminated after a finite number of iterations.

What remains to be done is to specify how the problems (17) and (18) and (19)–(22) can be solved computationally.

#### 4.1.1. Solution of (17) and (18)

Problem (17) is solved by branch-and-bound as follows.

**Branching strategy:** The “forbidden set branching strategy” (Stork, 2001) is used. With each node of the branch-and-bound tree, a partial solution represented by a selection  $X \subseteq (V \times V) \setminus A$  is associated. Selection  $X$  resolves some, but in general yet not all given MFSs. The forbidden set branching scheme arranges all MFSs in a fixed order and assigns the first level of the tree to the first MFS, the second level to the second MFS, etc. The node with associated partial solution  $X$  is split into successor nodes, where each successor is obtained from one particular additional precedence resolving the MFS corresponding to the current level: For MFS  $U$ , the set of these precedences is  $\{(i, j) \mid i, j \in U, i \neq j\}$ . As soon as the procedure has reached the level of leaves (the level below that corresponding to the last MFS), all MFSs have been resolved, such that a sufficient selection  $p$  representing a complete solution is obtained. The objective value of such a solution  $p$  in (17) can be computed directly. This produces an upper bound.

**Bound computation:** We look for a lower bound in a node corresponding to a subset  $X$  of the precedences. Let  $p$  be a sufficient selection associated with a leaf of the subtree rooted in the considered node. Then  $X \subseteq p$ . Let  $g^\mu(X, \mathbf{m})$  denote the length of a critical path in the graph with edge set  $T(A \cup X)$  under scenario  $\omega_\mu$  and mode combination  $\mathbf{m}$ . (This is consistent with the definition of the makespan  $g^\mu(p, \mathbf{m})$ .) Because of  $X \subseteq p$ , we have that  $g^\mu(X, \mathbf{m}) \leq g^\mu(p, \mathbf{m})$  for all  $\mu = 1, \dots, N$ . Therefore

$$\max_{(w,v) \in W^s} \left( \sum_{\mu=1}^N (\alpha^v + w\beta^v + g^\mu(X, \mathbf{m}) + w(\mathbf{c}^\mu)' \mathbf{m})_+ - \sum_{\mu=1}^N (\alpha^v + w\beta^v - \alpha^\mu - w\beta^\mu)_+ \right)$$

is a lower bound for the objective function in (17). If  $X = p$ , then this bound coincides with the objective function value. To reduce the computation time, it is possible to use a simple pruning strategy suggested in Stork (2001): Let  $U$  be the MFS that is used for branching on the node associated with set  $X$ . Whenever  $(i, j) \in T(A \cup X)$  for  $i \in U$  and  $j \in U$ , then it is optimal to resolve  $G$  by choosing the branch  $(i, j)$ , which does then not imply an extension of  $X$ .

#### 4.1.2. Solution of (19)–(22).

According to the solution approach in Homem-de Mello and Mehrotra (2009), problems (19)–(22) can be reformulated equivalently to the following binary mixed integer program (MIP):

$$\min_{w, y, e, h, z} \sum_{\mu=1}^N y_{\mu} - \sum_{\mu=1}^N e_{\mu} \quad (24)$$

$$\text{s.t. } a \leq w \leq b \quad (25)$$

$$y_{\mu} \geq \alpha^v + w\beta^v - \alpha^{\mu} - w\beta^{\mu} \quad (\mu = 1, \dots, N) \quad (26)$$

$$e_{\mu} - h_{\mu} = \alpha^v + w\beta^v - \hat{\alpha}^{\mu} - w\hat{\beta}^{\mu} \quad (\mu = 1, \dots, N) \quad (27)$$

$$Mz_{\mu} \geq e_{\mu} \quad (\mu = 1, \dots, N) \quad (28)$$

$$\begin{aligned} M(1 - z_{\mu}) &\geq h_{\mu} \quad (\mu = 1, \dots, N) \\ y_{\mu} &\geq 0, e_{\mu} \geq 0, h_{\mu} \geq 0 \quad (\mu = 1, \dots, N) \\ z_{\mu} &\in \{0, 1\} \quad (\mu = 1, \dots, N) \end{aligned} \quad (29)$$

with  $M$  a sufficiently large number. This is immediately seen, since constraints (28) and (29) ensure that either  $e_{\mu} = 0$  or  $h_{\mu} = 0$ , so (27) can only be satisfied if  $e_{\mu} = (\alpha^v + w\beta^v - \hat{\alpha}^{\mu} - w\hat{\beta}^{\mu})_+$  and  $h_{\mu} = (\alpha^v + w\beta^v - \hat{\alpha}^{\mu} - w\hat{\beta}^{\mu})_-$ . The above MIP contains  $N$  binary variables; if the number  $N$  of scenarios is not too large, the problem can be solved by a commercial MIP solver. (In our experiments, we use CPLEX for this purpose.)

#### 4.2. Solution of the subproblem

If, for a certain mode combination  $\mathbf{m}$ , the feasibility test described in the last subsection has produced the result that subproblem (13)–(15) possesses a feasible solution, we are left with the task of solving the optimization problem itself. For this purpose, an extension Algorithm 3 of Algorithm 2 can be applied. The idea is to minimize a weighted sum of the original objective function and a penalty term (obtained from the auxiliary objective function used in Algorithm 1) penalizing feasibility violations. The weight is adapted iteratively. We first present the algorithm and prove then its validity.

**Algorithm 3.** (Solution of the Subproblem).

1. Set  $s = 0$ ,  $w = a$ ,  $v = 1$ , and  $W^0 = \{(w, v)\}$ . Choose some  $\varepsilon > 0$ .
2. Solve the problem

$$\min_p \left\{ \epsilon \sum_{\mu=1}^N g^{\mu}(p, \mathbf{m}) + \max_{(w, v) \in W^s} \left[ \sum_{\mu=1}^N (\alpha^v + w\beta^v + g^{\mu}(p, \mathbf{m}) + w(\mathbf{c}^{\mu})' \mathbf{m})_+ - \sum_{\mu=1}^N (\alpha^v + w\beta^v - \alpha^{\mu} - w\beta^{\mu})_+ \right] \right\} \quad (30)$$

$$\text{s.t. } p \in \Pi_{\mathbf{m}} \quad (31)$$

Let  $\hat{p}$  denote the optimal solution.

3. Let  $\hat{\alpha}^{\mu} = -g^{\mu}(\hat{p}, \mathbf{m})$  and  $\hat{\beta}^{\mu} = -(\mathbf{c}^{\mu})' \mathbf{m}$ . For  $v = 1, 2, \dots, N$ , solve (19)–(22). As soon as for some  $v$  in this loop, a negative solution value is obtained, go to Step 4.

4. If in Step 3, the solution values have been nonnegative for all investigated  $v = 1, \dots, N$ , return  $\hat{p}$  and **stop**. Otherwise, let  $v$  be the index for which a negative solution value has been obtained, let  $(w, y)$  be the corresponding solution, let

$$\begin{aligned} \delta &= \sum_{\mu=1}^N (\alpha^v + w\beta^v - \hat{\alpha}^{\mu} - w\hat{\beta}^{\mu})_+ \\ &\quad - \sum_{\mu=1}^N (\alpha^v + w\beta^v - \alpha^{\mu} - w\beta^{\mu})_+ \geq -\psi_v(w, y) > 0, \end{aligned}$$

and set  $\varepsilon = \min(\varepsilon, \rho \delta / T_{\mathbf{m}})$ , where  $\rho$  is a constant with  $0 < \rho < 1$  and  $T_{\mathbf{m}}$  is an upper bound for  $\sum_{\mu=1}^N g^{\mu}(p, \mathbf{m})$  ( $p \in \Pi_{\mathbf{m}}$ ). Add  $w = w^s$  and  $v = v^s$  to  $W^s$  by setting  $W^s = W^s \cup \{(w^s, v^s)\}$ .

5. Set  $s = s + 1$  and go to Step 2.

Note that as (17), also (30) is a nondecreasing function of the makespan  $g^{\mu}(p, \mathbf{m})$ . Therefore, problem (30) and (31) can again be solved based on the forbidden set branching scheme, as described in Section 4.1.1.

**Proposition 1.** If the feasibility criterion (16) is satisfied, then Algorithm 3 terminates after a finite number of iterations with an optimal solution of (13)–(15).

The proofs of the propositions are given in Appendix A.

In order to reduce the runtime of Algorithm 1, the following bounds can be helpful:

**Proposition 2.** Let  $\bar{g}(p, \mathbf{m}) = (1/N) \sum_{v=1}^N g^v(p, \mathbf{m})$  and  $\bar{\mathbf{c}} = (1/N) \sum_{v=1}^N \mathbf{c}^v$ . Then  $(F_1(p, \mathbf{m}), F_2(\mathbf{m})) \succeq_{\text{lin}}^N (F_1(p_0, \mathbf{m}_0), F_2(\mathbf{m}_0))$  implies that for all  $w \in [a, b]$ :

$$(1/w) \bar{g}(p, \mathbf{m}) + \bar{\mathbf{c}}' \mathbf{m} \leq (1/w) \bar{g}(p_0, \mathbf{m}_0) + \bar{\mathbf{c}}' \mathbf{m}_0 =: C_w, \quad (32)$$

$$\bar{\mathbf{c}}' \mathbf{m} \leq \bar{\mathbf{c}}' \mathbf{m}_0 + (1/w) (\bar{g}(p_0, \mathbf{m}_0) - \bar{g}(\emptyset, \mathbf{m})). \quad (33)$$

Because of (32), only mode combinations  $\mathbf{m}$  for which  $\bar{\mathbf{c}}' \mathbf{m} \leq C_w$  have to be enumerated. For  $w = b$ , the best (i.e. lowest) upper bound  $C_w$  is obtained. For each  $\mathbf{m}$  satisfying the condition above, one can check then whether (33) is fulfilled. Therein, it is optimal to set  $w = b$  if  $\bar{g}(p_0, \mathbf{m}_0) \geq \bar{g}(p, \mathbf{m})$ , and to set  $w = a$  otherwise. Only if also (33) holds, it is necessary to execute Algorithms 2 and 3.

**Example 2.** The following very small example illustrates some basic ideas of the proposed algorithms. Consider  $n = 4$  activities with precedences  $A = \{(1, 2), (3, 4)\}$ . Let the durations and costs be given by

$$D = \begin{bmatrix} (5, 2, 7) & (2, 4, 6) \\ (3, 8, 4) & (1, 6, 5) \\ (1, 6, 5) & (3, 6, 7) \\ (4, 3, 2) & (8, 2, 1) \end{bmatrix} \quad \text{and} \quad C = \begin{bmatrix} (1, 2, 3) & (2, 4, 6) \\ (3, 1, 4) & (2, 1, 3) \\ (5, 5, 5) & (4, 4, 4) \\ (6, 7, 8) & (5, 8, 9) \end{bmatrix},$$

respectively. Therein, lines are activities, columns are modes, and the three consecutive values in the entries of  $D$  and  $C$  refer to  $N = 3$  scenarios. Furthermore, assume that there are three MFSs (the same in both modes), namely  $\{2, 3\}$ ,  $\{1, 2, 4\}$  and  $\{1, 3, 4\}$ . The two last MFSs are already resolved by  $A$ , so we can restrict ourselves to the sufficient selections  $\{(2, 3)\}$  and  $\{(3, 2)\}$ . As the reference solution, we choose the mode combination  $\mathbf{m}_0 = (2, 1, 1, 2)$  together with the sufficient selection  $p_0 = \{(2, 3)\}$ . The weight interval is chosen as  $[a, b] = [0.2, 1]$ .

Algorithm 1 starts by sorting the mode combinations according to their expected costs. Obviously, the mode combination with lowest expected cost is  $\mathbf{m} = (1, 2, 2, 1)$  with an expected cost value of 15. Therefore, in the first iteration of Algorithm 1, this mode combination is investigated. To test whether a solution to (8)–(11) exists,



Algorithm 2 is called. Algorithm 2 sets  $W^0 = \{(0.2, 1)\}$  and determines then the optimal sufficient selection  $\hat{p}$  w.r.t. (17), based on the values

$$\alpha^1 = -14, \alpha^2 = -20, \alpha^3 = -16, \beta^1 = -15, \beta^2 = -18, \beta^3 = -24,$$

which are the negative makespan and the negative cost values, respectively, of the reference solution in the three scenarios. (The last can easily be checked by observing that the reference solution executes the four activities in sequential order.) The best sufficient selection w.r.t. (17) turns out to be  $\hat{p} = \{(3, 2)\}$ , with assigned negative makespan and negative cost values of

$$\hat{\alpha}^1 = -7, \hat{\alpha}^2 = -12, \hat{\alpha}^3 = -12, \hat{\beta}^1 = -13, \hat{\beta}^2 = -14, \hat{\beta}^3 = -18.$$

This gives a solution value of  $-10.4 \leq 0$  in (17). The nonpositive solution value indicates that the found  $(\hat{p}, \mathbf{m})$  satisfies the condition for dominating the reference solution at least for  $w = 0.2$  and  $v = 1$ . Next, the program in Step 3 of Algorithm 2 tests whether for  $(\hat{p}, \mathbf{m})$ , there is any other  $w \in [a, b]$  or  $v \in \{1, \dots, N\}$  for which the dominance condition is not satisfied. If this should hold, the set of considered weights or scenarios would be extended. In our example, it is not the case, so Algorithm 2 decides that for mode combination  $\mathbf{m} = (1, 2, 2, 1)$ , a solution dominating the reference solution exists.

As a consequence, Algorithm 3 is called in order to maximally improve (for fixed  $\mathbf{m}$ ) the value of the expected makespan, if this is possible, keeping the dominance constraint satisfied. In our example, however, such an improvement cannot be achieved. Therefore, the found solution  $(\hat{p}, \mathbf{m})$  is inserted into the list of efficient solutions. Evidently, for its expected makespan and its expected cost, we obtain the values  $(7 + 12 + 12)/3 = 10.333$  and  $(13 + 14 + 18)/3 = 15$ , respectively. The next dominating solution occurs four iterations later for  $\mathbf{m} = (1, 2, 1, 1)$ . In this case, the optimal expected makespan results as 10, which is better than that of the previous solution. Therefore, also this solution is added to the list of efficient solutions, together with its expected cost value of 16. A third efficient solution is found for  $\mathbf{m} = (2, 2, 1, 1)$  (expected makespan 9.333, cost 18).

**Remark 2.** It is evident that in our approach, the stochastic dominance constraint does not assume the same preference structure as the final selection according to Pareto efficiency, since the former takes a risk-averse position, whereas the latter acts in a risk-neutral manner. Thus, at least from the perspective of expected utility theory (which is, however, not the only possible way to interpret stochastic dominance relations), the two steps of the decision process presume different types of utility functions. In this perspective, the proposed approach has to be considered as a pragmatic approximation to a more complex procedure.<sup>5</sup> It should be emphasized that also other well-established approaches in optimization, such as Chance-Constrained Programming, follow the pragmatic idea of filtering out, in a first step, all those solution that do not conform with a requirement  $A$  to a sufficiently high degree, and selecting then the final solution according to another requirement  $B$ , without considering  $A$  anymore.

## 5. Numerical examples

To test the described approach, two sets of instances were generated. Since already the single-objective deterministic MRCPSP and

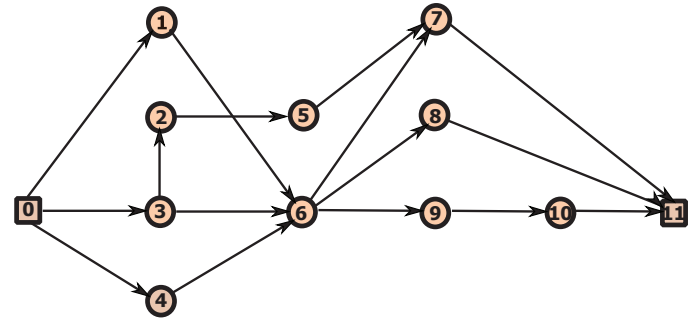


Fig. 1. Precedences graph  $G(V, A)$  for the smaller test instances.

the single-objective, single-mode S-RCPSP, which are both special cases of the RAMOS-MRCPSPP considered here, are computationally challenging problems, it is obvious that we can only hope to solve small instances of the RAMOS-MRCPSPP to optimality. For both sets of instances, a random test instance generator described below was used. The structure of the precedence graph  $G(V, A)$  was fixed in advance in both cases. The smaller instances have  $n = 10$  activities and are built on the graph in Fig. 1. To get somewhat larger instances, the first MRCPSP instance from the well-known PSPLIB (Kolisch & Sprecher, 1997) was taken. It contains  $n = 16$  non-dummy activities. Again by varying the concrete parameters with the help of the test instance generator, the second set of instances was created, building on the graph  $G(V, A)$  from the PSPLIB example case.

### 5.1. Test instance generation

For activity durations  $D_{im}$  as well as for costs  $C_{im}$ , uniform distributions on certain intervals  $[D_{im}, \bar{D}_{im}]$  and  $[C_{im}, \bar{C}_{im}]$ , respectively, were considered. In each test instance, the intervals assigned to each activity and each mode were generated randomly, as indicated below. A basic mode 1, a cheap mode 2 and an expensive mode 3 were distinguished. From the set of  $n$  activities, disjoint subsets  $N_{Ch}$ ,  $N_{Ex}$  and  $N_{ChEx}$  with pre-defined cardinalities  $n_{Ch}$ ,  $n_{Ex}$  and  $n_{ChEx}$  were randomly selected. Activities in  $N_{Ch}$  can be executed in modes 1 or 2, activities in  $N_{Ex}$  can be executed in modes 1 or 3, and activities in  $N_{ChEx}$  can be executed in modes 1, 2 or 3. The remaining activities can only be executed in the basic mode 1.

The generation of the intervals  $[D_{im}, \bar{D}_{im}]$  was controlled by pre-defined parameters  $\underline{D}_M, \bar{D}_M, \underline{D}_L, \bar{D}_L, \underline{C}_M, \bar{C}_M, \underline{C}_L, \bar{C}_L$ . In the basic mode, with  $\mathcal{U}(a, b)$  denoting a random number uniformly selected from  $[a, b]$ , we chose

$$\underline{D}_{i1} = \mathcal{U}(\underline{D}_M, \bar{D}_M), \quad \bar{D}_{i1} = \underline{D}_{i1} + \mathcal{U}(\underline{D}_L, \bar{D}_L),$$

$$\underline{C}_{i1} = \mathcal{U}(\underline{C}_M, \bar{C}_M), \quad \bar{C}_{i1} = \underline{C}_{i1} + \mathcal{U}(\underline{C}_L, \bar{C}_L).$$

The intervals for the cheap and for the expensive mode were generated from the intervals of the basic mode by the following transformations: For the cheap mode,

- $[\underline{D}_{i1}, \bar{D}_{i1}]$  was replaced by  $[\theta \chi \underline{D}_{i1}, \theta(\bar{D}_{i1} + (1 - \chi) \underline{D}_{i1})]$  with  $\theta = \mathcal{U}(1, \kappa_1)$  and  $\chi = \mathcal{U}(0, 1)$ , and
- $[\underline{C}_{i1}, \bar{C}_{i1}]$  was replaced by  $[\gamma \underline{C}_{i1}, \gamma \bar{C}_{i1}]$  with  $\gamma = \mathcal{U}(\kappa_2, 1)$ .

Thus, in the cheap mode, the durations tend to increase in mean value and variance, and the costs tend to decrease. For the expensive mode, the intervals were transformed in an analogous, but converse way.

A single resource type was assumed. The resource requirements  $r_{i1}$  of the activities were generated from a uniform distribution on an interval  $[R, \bar{R}]$ .

The reference solution  $(p_0, \mathbf{m}_0)$  for a test instance was determined as follows: The mode combination  $\mathbf{m}_0$  was obtained by taking only

<sup>5</sup> This procedure could work as follows: First, the proposed algorithm has to be executed in a modified variant where it produces all solutions satisfying the stochastic dominance constraint (such a modification is possible, but at the price of a considerable runtime increase). Then, to the resulting candidate set of solutions, some multi-criteria decision making method determining nondominated solutions with respect to a set  $\mathcal{U}$  of possible utility functions has to be applied. This could, for example, be Robust Ordinal Regression (Branke, Greco, Slowinski, & Zielniewicz, 2009). For  $\mathcal{U}$ , the set of utility functions generating the stochastic dominance relation  $\succeq_p^{lin}$  has to be taken. Although this procedure would be ideal from a theoretical point of view, its runtime requirements are certainly extremely high.

the basic mode 1 for all activities. The sufficient selection  $p_0$  was obtained by generating only a single scenario  $\omega_1$  and by solving the resulting deterministic RCPSP, i.e., by determining the sufficient selection with minimal makespan to the given  $\mathbf{m}$  and  $\omega_1$ .

To denote the test instances and their variants, we use in the following the code  $n/a_1/[a, b]/\text{seed}$ , containing number of activities, availability of the resource, endpoints of the interval restricting the relative weight of the second objective, and used seed for the random generation of the instance.

### 5.2. Results for instances with $n = 10$

The following parameter values for the instance generation were used:  $n_{Ch} = 2$ ,  $n_{Ex} = 2$ ,  $n_{ChEx} = 3$  (this gives  $2^2 \cdot 2^2 \cdot 3^3 = 432$  mode combinations),  $[D_M, \bar{D}_M] = [5, 20]$ ,  $[D_L, \bar{D}_L] = [0, 30]$ ,  $[C_M, \bar{C}_M] = [5, 20]$ ,  $[C_L, \bar{C}_L] = [0, 30]$ ,  $[R, \bar{R}] = [1, 5]$ ,  $\kappa_1 = 2$ ,  $\kappa_2 = 0.5$ . For  $a_1$ , the values 5, 6 and 7 were tested, and for the interval  $[a, b]$  of the weights, the alternatives  $[a, b] = [0.1, 1.0]$  and  $[a, b] = [0.5, 0.7]$  were compared. On a PC with 2.4 gigahertz Quad Core Processor and 8 GB System Memory, a MATLAB R2007b implementation required 8.5 minutes computation time per test instance in the average.

Fig. 2 shows the obtained Pareto frontiers for some selected instances. Let us give some comments. Typically, the Pareto frontiers contained comparably few elements. Interestingly, in many cases the covered ranges of expected makespan and expected cost were nevertheless rather large (see, e.g., the two plots in the upper row of Fig. 2). However, also cases occurred (see, e.g., the left plot in the medium row of Fig. 2) where one of the objectives covered only a small range. Of course, these properties depend on type and quality of the reference solution.

The influence of the interval  $[a, b]$  of the weight  $w$  of the cost objective in the stochastic dominance relation is seen by a comparison of the left and the right plots in the first two rows of Fig. 2. In the upper row (instance 11), we see that a reduction of this interval from  $[0.1, 1.0]$  to  $[0.5, 0.7]$  simply extends the Pareto frontier by adding three points at the right end. It is clear that a restriction of the interval  $[a, b]$ , which means that the decision maker can estimate the relative importance of the two objectives in a more specific way, will increase the chance that a solution dominates the reference solution and will thus tend also to “enlarge” the Pareto frontier, but the concrete form of this extension cannot be predicted. In the medium row of Fig. 2 (instance 13), the restriction of the interval  $[a, b]$  has the consequence that two new nondominated points at the left end of the Pareto frontier arise, the second of which displaces the previous leftmost point from the frontier.

The two plots in the bottom row of Fig. 2 illustrate the influence of the resource availability: Compared to the first plot in the upper row,  $10/6/[0.1, 1.0]/11$ , we reduce here the availability  $a_1$  from 6 to 5 (left plot) and increase it to 7 (right plot). It can be seen that the shape of the frontier remains roughly similar, but of course the reduction of  $a_1$  tends to increase the makespan, whereas the enlargement of  $a_1$  tends to decrease it.

### 5.3. Results for instances with $n = 16$

Now we used the following parameter values for the instance generation:  $n_{Ch} = 2$ ,  $n_{Ex} = 2$ ,  $n_{ChEx} = 2$ ,  $[D_M, \bar{D}_M] = [10, 20]$ ,  $[D_L, \bar{D}_L] = [0, 30]$ ,  $[C_M, \bar{C}_M] = [5, 20]$ ,  $[C_L, \bar{C}_L] = [0, 30]$ ,  $[R, \bar{R}] = [1, 5]$ ,  $\kappa_1 = 2$ ,  $\kappa_2 = 0.5$ . For  $a_1$  and for the interval  $[a, b]$  of the weights, the same choices were tested as in the case of the smaller instances. The average computation time per test instance increased now to 2.6 hours.

The obtained results were qualitatively similar to those of the smaller instances. Fig. 3 shows two Pareto frontiers as an example. As it can be seen, the frontiers contain few elements, but the covered ranges of the objective functions are large enough for making the differences of practical relevance. Again, the reduction of the interval

$[a, b]$  extends the Pareto frontier, in this case now by adding a new point at the right.

Admittedly, instances with  $n = 16$  activities are still rather small. However, let us mention that in the area of stochastic real-time systems, to which the approach presented here is applicable in principle, instances of this size can already be realistic. (For related problems in this area, cf. Manolache, Eles, and Peng, 2001; Mills and Anderson, 2011; Tidwell, Bass, Lasker, Wylde, Gill, and Smart, 2011.) It should be emphasized that although the computation of the optimal ES policy for 16 jobs takes about 2 hours, the execution of the found ES policy only requires milliseconds of computation time and can therefore also be integrated in a real-time system with very restrictive time limits.

As mentioned in Remark 1, it is possible that the reference solution is already very good, with the consequence that a too small Pareto frontier results. (In the extreme case, the Pareto frontier may even degenerate to the reference solution itself.) In the literature on stochastic dominance constraints, a conceptually simple way to overcome this problem has been proposed, namely to decrease the quality of the reference solution in a controlled way in order to increase the number of efficient solutions (see, e.g., Armbruster and Luedtke, 2011). In the method proposed here, this can be done by using a multiplicative tolerance level, applied to the reference solution. For this purpose, we replace in (7) the outcome vector  $\mathbf{Y}$  of the reference solution (with components negative makespan and negative cost) by the vector  $\Gamma \cdot \mathbf{Y}$  where  $\Gamma > 1$ . An example of the effect is shown in Fig. 4, which is based on the same test instance as Fig. 3, left plot: We see that by allowing a 5 percent tolerance ( $\Gamma = 1.05$ ), the cardinality of Pareto frontier increases from 4 to 6, and by a 20 percent tolerance ( $\Gamma = 1.2$ ), it further increases to 8.

### 5.4. Results for larger instances

As seen from the computation times reported below for  $n = 16$ , there is little hope that instance sizes beyond  $n = 20$  can be solved exactly. Heuristics or metaheuristics will be needed to cope with medium-sized or large instances. A development and comprehensive test of a bi-objective metaheuristic for the problem under consideration is outside the scope of this paper. However, to illustrate the applicability of the proposed approach also for somewhat larger instances, the presented algorithms have been modified by solving problems (17) and (18) and (30) and (31) only heuristically. For this purpose, a Variable Neighborhood Search (VNS) algorithm has been implemented. (We omit here the implementation details for the sake of brevity, mentioning only that for the “move-or-not” decision, a Simulated-Annealing-based acceptance rule has been used, as in many modern VNS implementations.) Evidently, for those mode combinations  $\mathbf{m}$  for which a feasible sufficient selection can be found in this way, the solution value produced by the heuristic modification of Algorithm 3 yields an upper bound for the true solution value of (13) and (15).

To evaluate the quality of the solution, we compute also a lower bound. It can be shown by a short derivation that each lower bound on the solution value of (30) and (31), divided by  $\varepsilon$ , yields also a lower bound for (13)–(15). A lower bound of the former kind, however, can easily be obtained by terminating the branch-and-bound procedure for the solution of (30) and (31) already after a certain number of iterations. The resulting gap to the upper bound produced by the VNS measures the solution quality achieved for the current mode combination  $\mathbf{m}$ . Let us emphasize, however, that even if the gap vanishes, we have no guarantee that the current solution is Pareto-efficient: Since also Algorithm 2 is solved now only heuristically, it could happen that some mode combination is not recognized as possessing a feasible solution, and such a “lost” solution could dominate the current solution. Therefore, the computed gaps provide only a partial assessment of the quality of the Pareto frontier; in a strict sense, they

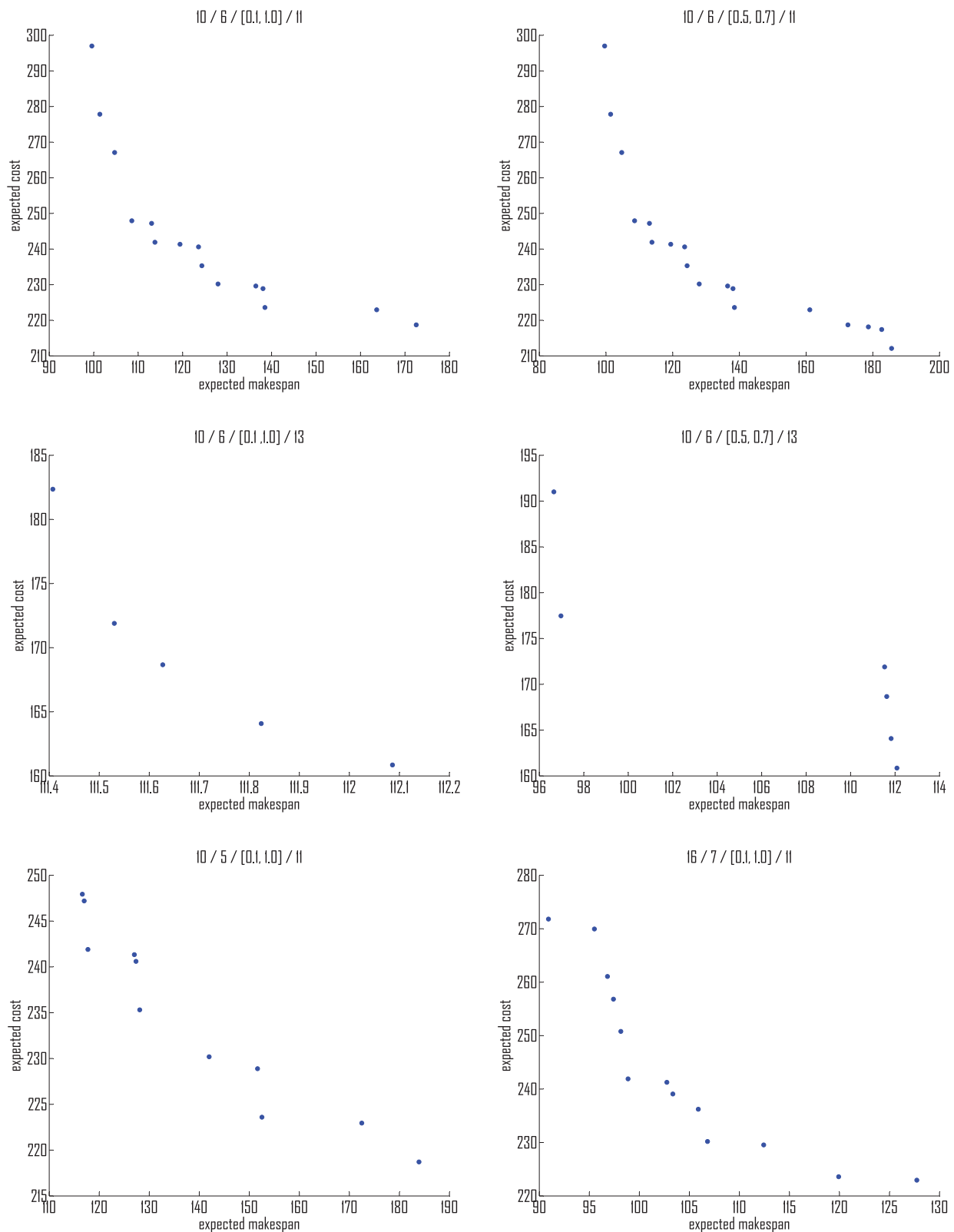


Fig. 2. Some Pareto frontiers for instances with  $n = 10$ .

only apply to each cost level (defined by a mode combination) in separation from the others.

The parameters of the VNS procedure were tuned in such a way that the computation times for the test instances of size  $n = 16$  were in the same order of magnitude as those required for the exact

solution. Nevertheless, it turned out that already for these generous computation times, the heuristic modification did usually not provide the exact Pareto frontier. In the instance corresponding to the left plot of Fig. 3, all four mode combinations containing Pareto optima were found, but in two of the four cases, the corresponding sufficient



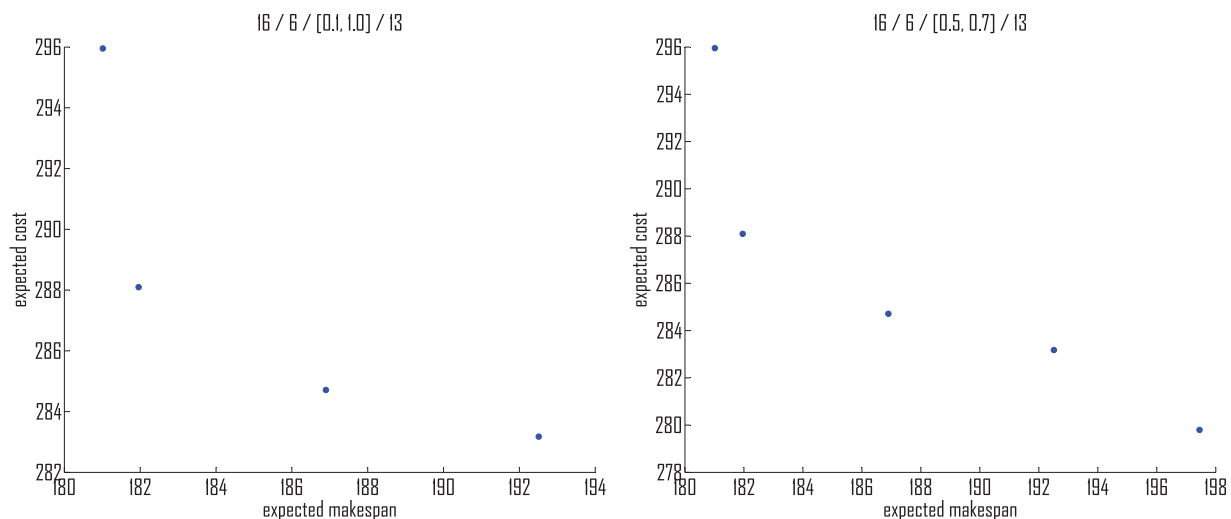


Fig. 3. Some Pareto frontiers for instances with  $n = 16$ .

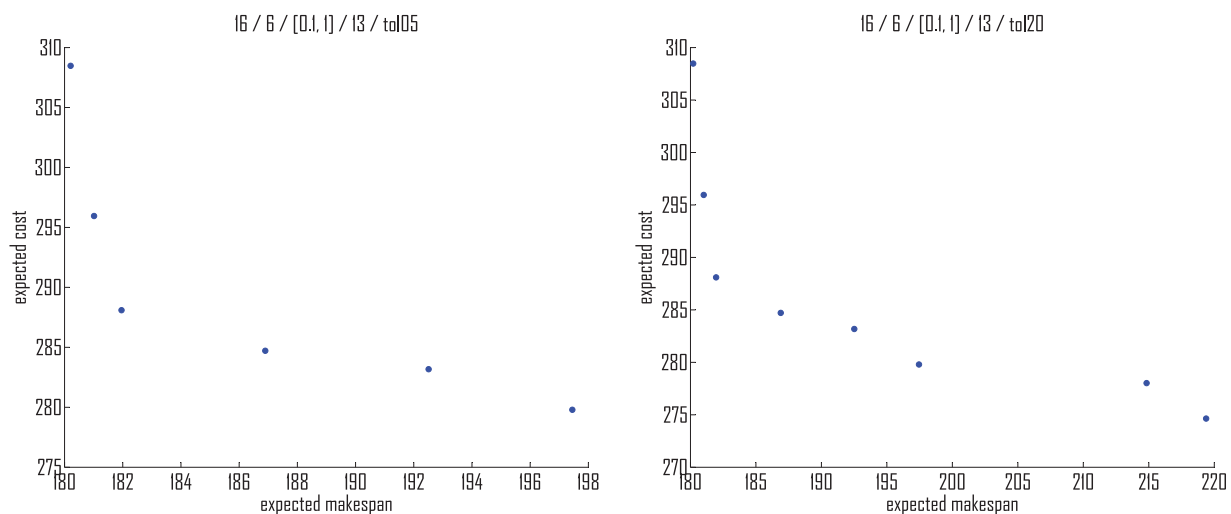


Fig. 4. Pareto frontiers for the instance of Fig. 3, left plot, after application of tolerance level  $\Gamma = 1.05$  (left) and  $\Gamma = 1.20$  (right).

selections were slightly suboptimal (makespan gaps of 1.18 percent and 0.80 percent, respectively). In the instance corresponding to the right plot, four of the five Pareto optima were reproduced exactly, but one was missed.

Now, two larger test instances were generated: First, the graph of Fig. 1, serving as the precedence graph for the  $n = 10$  instances, was concatenated with the precedence graph for the  $n = 16$  instances taken from the PSPLIB test case by turning the terminal node of the first graph into a non-dummy node and appending the second graph to it. The produced graph for  $n = 27$  activities was extended to a problem instance by the test instance generator described in the beginning of this section, using  $[\underline{R}, \bar{R}] = [1, 7]$ ,  $a_1 = 7$ ,  $[a, b] = [0.5, 0.7]$  and all other parameter values as for the  $n = 16$  instances. We call this instance n27. Second, the last MRCPSP test case of size  $n = 30$  contained in the PSPLIB was taken and extended to a test instance n30 for our problem by the same generation procedure (with the same parameters) as for n27.

The Pareto frontiers produced by the heuristic algorithm for n27 and n30 are shown in Tables 1 and 2 (Appendix B), respectively. It can be seen that for n27, very small gaps (hardly exceeding 1 percent) are obtained. For n30, the gaps are considerably larger (around 30 percent), which is probably due to the much higher degree of parallelism

in the precedence graph of n30 compared to n27. However, it is important to note that the large gaps for n30 do not necessarily mean that the produced heuristic solutions are poor. There are indications for the conjecture that the computed lower bounds are rather weak in this case, such that the true optimum is nearer to the upper than to the lower bound. In this context, it is also interesting to compare the dominance-constrained problem with its relaxation obtained by dropping the stochastic dominance constraints. For n30, the gap between the (found) expected makespan of a proposed-efficient solution of the dominance-constrained problem and the (found) expected makespan after relaxing is only around 4 percent in the average. On the other hand, the lower bound obtained for the relaxed problem is usually also almost 30 percent below the found solution value. This suggests that the solution quality achieved by the presented approach is not essentially worse than that achieved by heuristic solutions of the simpler expected makespan minimization problem.

## 6. Conclusions

Real-life project scheduling is a complex process that can often not be adequately represented by standard models from the project

scheduling literature. One of the drawbacks can be that these models typically do not take uncertainty and risk aversion, multiple available execution modes, and multiple objectives *simultaneously* into account. The present paper provides a first model integrating all these features, as well as an exact solution technique for the determination of the Pareto frontier. The approach relies on optimization under multivariate stochastic dominance constraints and provides also a prototype for the application of this method in connection with Pareto analysis. A branch-and-bound technique for the optimization of early-start policies developed in the stochastic project scheduling literature has been used as an essential solution component. Although the instances that could be solved to optimality are comparably small, they show the principal viability of the approach.

The next natural step of research will be the development of heuristic solution techniques able to cope with large problem instances. Multiobjective evolutionary algorithms such as the NSGA-II, combined with mathematical programming routines in a “matheuristic” fashion, are obvious candidates for solution approaches of this kind. In this context, the solution method proposed in this paper will be valuable insofar as it can provide, for comparison purposes, a benchmark of instances whose exact solutions are known. Another topic future research should address is the situation where instead of early-start policies, other classes of stochastic scheduling policies such as activity-based priority policies, resource-based priority policies or preprocessor policies are used. Furthermore, as an alternative to the optimization under stochastic dominance constraints, also other risk-averse decision making approaches, e.g., the use of the conditional value at risk (CVaR), should be investigated.

## Appendix A

**Proof of Proposition 1.** First we show that the algorithm terminates after a finite number of steps. Let  $\hat{p}_s$  denote the solution  $\hat{p}$  of iteration  $s$ , and let  $\delta_s$  and  $\varepsilon_s$  denote the values of  $\delta$  and  $\varepsilon$  as determined in iteration  $s$ , respectively. For abbreviation, set

$$\Phi_{w,v}(p) = \left[ \sum_{\mu=1}^N (\alpha^\nu + w\beta^\nu + g^\mu(p, \mathbf{m}) + w(\mathbf{c}^\mu)' \mathbf{m})_+ - \sum_{\mu=1}^N (\alpha^\nu + w\beta^\nu - \alpha^\mu - w\beta^\mu)_+ \right]_+.$$

If the algorithm does not stop in iteration  $s$ , then  $\delta_s$ ,  $w^s$  and  $v^s$  are defined, and  $\Phi_{w^s, v^s}(\hat{p}_s) = (\delta_s)_+ > 0$ . Moreover, the solution  $\hat{p}_s$  cannot solve (30) and (31) in any later iteration  $s' > s$ . This is seen as follows: By the assumption that (16) holds, there is a feasible solution  $p^*$  with  $\max_{(w,v) \in W^s} \Phi_{w,v}(p^*) = 0$  for all  $s$ . For  $s' > s$ , the objective value

of  $p^*$  in (30) becomes

$$\varepsilon_{s'} \sum_{\mu=1}^N g^\mu(p^*, \mathbf{m}) + \max_{(w,v) \in W^{s'}} \Phi_{w,v}(p^*) \leq \varepsilon_{s+1} T_{\mathbf{m}} < \delta_s$$

in view of the update equation of the  $\varepsilon$  values. On the other hand, recalling that  $W^0 \subseteq W^1 \subseteq \dots$ , one obtains

$$\begin{aligned} \varepsilon_{s'} \sum_{\mu=1}^N g^\mu(\hat{p}_s, \mathbf{m}) + \max_{(w,v) \in W^{s'}} \Phi_{w,v}(\hat{p}_s) \\ \geq \max_{(w,v) \in W^{s+1}} \Phi_{w,v}(\hat{p}_s) \geq \Phi_{w^s, v^s}(\hat{p}_s) \geq \delta_s, \end{aligned}$$

and therefore  $\hat{p}_s$  cannot be optimal in iteration  $s'$ . We conclude that the solutions  $\hat{p}_s$  ( $s = 0, 1, \dots$ ) are pairwise different. Therefore, since all  $\hat{p}_s$  are contained in  $\Pi_{\mathbf{m}}$  and  $\Pi_{\mathbf{m}}$  is finite, the algorithm must terminate after a finite number of iterations.

For the solution  $\hat{p}_{s^*}$  obtained in the last iteration  $s^*$ , solving (19)–(22) in Step 3 of Algorithm 3 does not find a pair  $(w, v)$  disproving the claim that  $(\hat{p}_{s^*}, \mathbf{m})$  stochastically dominates  $(p_0, \mathbf{m}_0)$ , so  $\hat{p}_{s^*}$  is feasible for (13)–(15) (cf. the comments to Step 3 of Algorithm 2). In particular,

$$\max\{\Phi_{w,v}(\hat{p}_{s^*}) \mid a \leq w \leq b, v = 1, \dots, N\} = 0,$$

and hence the “max” term in (30) vanishes. For any other solution  $p'$  that is feasible w.r.t. (13)–(15), this term vanishes as well. So the assumption that  $\sum_{\mu=1}^N g^\mu(p', \mathbf{m}) < \sum_{\mu=1}^N g^\mu(\hat{p}_{s^*}, \mathbf{m})$  would contradict the optimality of  $\hat{p}_{s^*}$  for (30) in iteration  $s^*$ . This shows the optimality of  $\hat{p}_{s^*}$  for (13).  $\square$

**Proof of Proposition 2.** By definition, the stochastic dominance condition in the proposition is equivalent to

$$F_1(p, \mathbf{m}) + w F_2(\mathbf{m}) \succeq_{(2)} F_1(p_0, \mathbf{m}_0) + w F_2(\mathbf{m}_0) \quad \forall w \in [a, b].$$

Since second-order stochastic dominance implies superiority w.r.t. the expected values, it follows that

$$\begin{aligned} \mathbf{E}[F_1(p, \mathbf{m})] + w \mathbf{E}[F_2(\mathbf{m})] \\ \geq \mathbf{E}[F_1(p_0, \mathbf{m}_0)] + w \mathbf{E}[F_2(\mathbf{m}_0)] \quad \forall w \in [a, b]. \end{aligned}$$

Substitution  $\mathbf{E}[F_1(p, \mathbf{m})] = -\bar{g}(p, \mathbf{m})$  and  $\mathbf{E}[F_2(\mathbf{m})] = -\bar{\mathbf{c}}' \mathbf{m}$  and division by  $w$  yields (32).

As to (33), note that  $\bar{g}(p, \mathbf{m}) \geq \bar{g}(\emptyset, \mathbf{m})$  because of  $\emptyset \subseteq p$ . Therefore, (32) gives

$$\bar{\mathbf{c}}' \mathbf{m} \leq C_w - (1/w) \bar{g}(\emptyset, \mathbf{m}),$$

hence (33) follows by insertion.  $\square$

## Appendix B

**Table 1**

Approximated Pareto frontier of test instance n27 and lower bounds for the makespan values. Runtime: 3.8 hours. Gaps express the ratio (found value – lower bound)/(found value) in percent.

E[makespan]	399.010	383.939	377.038	374.354	369.446	363.322
E[cost]	560.021	562.399	574.364	574.537	586.503	603.388
Lower bound E[makespan]	396.875	383.939	377.038	371.084	364.549	358.424
Gap (in percent)	0.54	0	0	0.87	1.33	1.35

**Table 2**

Approximated Pareto frontier of test instance n30 and lower bounds for the makespan values. Runtime: 16.8 hours. Gaps as in Table 1.

E[makespan]	465.650	451.382	449.702	439.729	435.723	428.336	419.190	410.605	398.913	391.697
E[cost]	576.705	580.833	582.893	583.452	589.760	590.310	594.889	610.675	616.881	632.667
Lower bound E[makespan]	330.418	301.690	298.263	301.195	300.061	300.406	293.947	293.670	295.381	294.909
Gap (in percent)	29.04	33.16	33.68	31.50	31.13	29.87	29.88	28.48	25.95	24.71

## References

- Abbasi, B., Shadrokh, S., & Arkat, J. (2006). Bi-objective resource-constrained project scheduling with robustness and makespan criteria. *Applied Mathematics and Computation*, 180, 146–152.
- Armbruster, B., Luedtke, J. B. (2011). Models and formulations for multivariate dominance constrained stochastic programs. [http://www.optimization-online.org/DB\\_FILE/2010/05/2613.pdf](http://www.optimization-online.org/DB_FILE/2010/05/2613.pdf)
- Artigues, C., Leus, R., & Nobibon, T. (2013). Robust optimization for resource-constrained project scheduling with uncertain activity durations. *Flexible Services and Manufacturing Journal*, 25, 175–205.
- Ashtiani, B., Leus, R., & Aryanezhad, M.-B. (2011). New competitive results for the stochastic resource-constrained project scheduling problem: Exploring the benefits of pre-processing. *Journal of Scheduling*, 14, 157–171.
- Ballestín, F. (2007). When it is worthwhile to work with the stochastic RCPSp? *Journal of Scheduling*, 10, 153–166.
- Ballestín, F., & Leus, R. (2009). Resource-constrained project scheduling for timely project completion with stochastic activity durations. *Production and Operations Management*, 18, 459–474.
- Bouleimen, K., & Lecocq, H. (2003). A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version. *European Journal of Operational Research*, 149, 268–281.
- Branke, J., Greco, S., Slowinski, R., & Zielniewicz, P. (2009). Interactive evolutionary multiobjective optimization using robust ordinal regression. In *Evolutionary multi-criterion optimization* (pp. 554–568). Springer Berlin Heidelberg.
- Brucker, P., Drexler, A., Möhring, R., Neumann, K., & Pesch, E. (1999). Resource-constrained project scheduling: Notation, classification, models, and methods. *European Journal of Operational Research*, 112, 3–41.
- Chen, W. N., Zhang, J., Liu, O., & Liu, H. L. (2010). A Monte-Carlo ant colony system for scheduling multi-mode projects with uncertainties to optimize cash flows. In *Proc. evolutionary computation (CEC)* (pp. 1–8). Barcelona: 2010 IEEE Congress on Evolutionary Computation.
- Coelho, J., & Vanhoucke, M. (2011). Multi-mode resource-constrained project scheduling using RCPSp and SAT solvers. *European Journal of Operational Research*, 213, 73–82.
- Deblaere, F., Demeulemeester, H., & Herroelen, W. (2011). Proactive policies for the stochastic resource-constrained project scheduling problem. *European Journal of Operational Research*, 214, 308–316.
- Deblaere, F., Demeulemeester, H., & Herroelen, W. (2011). Reactive scheduling in the multi-mode RCPSp. *Computers & Operations Research*, 38, 63–74.
- Dentcheva, D., & Ruszczyński, A. (2009). Optimization with multivariate stochastic dominance constraints. *Mathematical Programming (Ser. B)*, 117, 111–127.
- Godinho, P., & Branco, F. G. (2012). Adaptive policies for multi-mode project scheduling under uncertainty. *European Journal of Operational Research*, 216, 553–562.
- Gutjahr, W. J., & Pichler, A. (2013). Stochastic multi-objective optimization: A survey on non-scalarizing methods. *Annals of Operations Research*. doi:10.1007/s10479-013-1369-5.
- Gutjahr, W. J., Strauss, C., & Wagner, E. (2000). A stochastic branch-and-bound approach to activity crashing in project management. *INFORMS Journal on Computing*, 12, 125–135.
- Hartmann, S., & Briskorn, D. (2010). A survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of Operational Research*, 207, 1–14.
- Herroelen, W., & Leus, R. (2005). Project scheduling under uncertainty: Survey and research potentials. *European Journal of Operational Research*, 165, 289–306.
- Homem-de Mello, T., & Mehrotra, S. (2009). A cutting-surface method for uncertain linear programs with polyhedral stochastic dominance constraints. *SIAM Journal on Optimization*, 20, 1250–1273.
- Igelmund, G., & Radermacher, F. J. (1983). Preselective strategies for the optimization of stochastic project networks under resource constraints. *Networks*, 13, 1–28.
- Jozefowska, J., Mika, M., Rozycki, R., Waligora, G., & Weglarz, J. (2001). Simulated annealing for multi-mode resource-constrained project scheduling. *Annals of Operations Research*, 102, 1–4.
- Kilic, M., Ulusoy, G., & Serifoglu, F. S. (2008). A bi-objective genetic algorithm approach to risk mitigation in project scheduling. *International Journal of Production Economics*, 112, 202–216.
- Kolisch, R., & Sprecher, A. (1997). PSPLIB – A project scheduling problem library. *European Journal of Operational Research*, 96, 205–216.
- Laumanns, M., Thiele, L., & Zitzler, E. (2006). An efficient, adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method. *European Journal of Operational Research*, 169, 932–942.
- Leus, R. (2011). Resource allocation by means of project networks: Dominance results. *Networks*, 58, 50–58.
- Leus, R., & Herroelen, W. (2004). Stability and resource allocation in project planning. *IEEE Transactions*, 36, 667–682.
- Manolache, S., Eles, P., & Peng, Z. (2001). Memory and time-efficient schedulability analysis of task sets with stochastic execution time. In *13th Euromicro conference on real-time systems* (pp. 19–26). IEEE.
- Mills, A. F., & Anderson, J. H. (2011). A multiprocessor server-based scheduler for soft real-time tasks with stochastic execution demand. *2011 IEEE 17th international conference on embedded and real-time computing systems and applications (RTCSA)*, 1, 207–217.
- Möhring, R. H., & Stork, F. (2000). Linear preselective policies for stochastic project scheduling. *Mathematical Methods of Operations Research*, 52, 501–515.
- Müller, A., & Stoyan, D. (2002). *Comparison methods for stochastic models and risks*. Chichester: John Wiley and Sons.
- Muller, L. F. (2011). A multi-mode resource-constrained project scheduling problem with stochastic nonrenewable resource consumption. *Technical Report*. DTU Management Engineering, Lyngby, Denmark.
- Roman, D., Darby-Dowman, K., & Mitra, G. (2006). Portfolio construction based on stochastic dominance and target return distributions. *Mathematical Programming (Ser. B)*, 108, 541–569.
- Stork, F. (2001). *Stochastic resource-constrained project scheduling*. Technische Universität Berlin (PhD thesis).
- Tereso, A. P., Araujo, M. T., & Elmaghraby, S. (2004). Adaptive resource allocation in multimodal activity networks. *International Journal of Production Economics*, 92, 1–10.
- Tereso, A. P., Araujo, M. T., Moutinho, R. S., Elmaghraby, S., et al. (2008). Project management: Multiple resources allocation. In *Proc. international conference on engineering optimization*. Rio de Janeiro, ISBN 978-85-7650-156-5.
- Tidwell, T., Bass, C., Lasker, E., Wylde, M., Gill, C. D., & Smart, W. D. (2011). Scalable utility aware scheduling heuristics for real-time tasks with stochastic non-preemptive execution intervals. *2011 23rd Euromicro conference on real-time systems (ECRTS)*, 238–247.
- Zhu, G., Bard, J., & Yu, G. (2007). A two-stage stochastic programming approach for project planning with uncertain activity durations. *Journal of Scheduling*, 10, 167–180.