The proactive and reactive resource-constrained project scheduling problem

Davari M, Demeulemeester E.

The proactive and reactive resource-constrained project scheduling problem

Morteza Davari^{†*} and Erik Demeulemeester[†]

Abstract

Uncertainty has become an inevitable aspect of project scheduling. We study the resource-constrained project scheduling problem (RCPSP) with stochastic durations. One of the most studied approaches to deal with stochastic durations is that of proactive and reactive scheduling. In this paper, we formulate an integrated proactive and reactive scheduling problem with a combined cost function which includes a baseline schedule cost as well as costs of a series of reactions. We propose four dynamic programming based models (Models 1-4) that solve the problem until optimality over different classes of policies. We compare our models with each other and with a combination of a traditional proactive solution (STC) and a reactive solution (RP_SGS). Computational results show that Model 2 outperforms the traditional solution only when reaction costs are greater than zero. Moreover, Model 3 and Model 4 clearly outperform Model 1 and Model 2 in all settings and the traditional solution in most of the settings.

Keywords: RCPSP; Stochastic activity durations; Proactive and reactive scheduling; Markov decision process.

1 Introduction

Until recently, the research on scheduling and project scheduling has mostly considered problems with fixed resource capacities and deterministic durations. In real-world environments, however, it is unrealistic to obtain only deterministic information. Even if we somehow manage to obtain estimated deterministic information, it is very difficult to rely on this information due to internal and/or external unexpected events. Therefore, uncertainty has become an inevitable aspect of project scheduling in the recent decades.

In project scheduling, handling uncertainty is of much greater importance than in other areas in scheduling since a project is a huge and costly one-time action. Each year, numerous projects go over budget or drag on long after their planned completion times (Flyvbjerg et al., 2003). Many project scheduling problems dealing with uncertainty have been explored and discussed by several authors during the last few decades (Demeulemeester and Herroelen, 2011).

When it comes to dealing with uncertainty in project scheduling, two main strands of research received lots of attention: the research on the stochastic RCPSP (SRCPSP) (Stork, 2001) and the research on proactive and reactive project scheduling (Leus, 2003). In the following paragraphs, we briefly discuss the state of the art in these two main strands.

The first body of research focuses on the SRCPSP which considers stochastic activity durations while minimizing the makespan. In the SRCPSP, solutions are *policies* rather than schedules because activity durations are not known in advance and thus activity starting times cannot be decided. A policy is a set of decision rules that dictate certain actions at certain decision moments, these being the finishing times of the activities. At each decision moment, we decide to start or

[†]Department of Decision Sciences and Information Management, Faculty of Business and Economics, KU Leuven (Belgium), firstname.lastname@kuleuven.be

*Corresponding Author

not to start a set of resource- and precedence-feasible activities. It is worth mentioning that the SRCPSP has been a target of criticism mainly because it does not provide a baseline schedule (Herroelen, 2005) and thus lacks robustness (Lamas and Demeulemeester, 2015).

To the best of our knowledge, the paper written by Igelmund and Radermacher (1983) is the first mainstream international English-language publication studying the SRCPSP. Although the SRCPSP has been known for several decades, Creemers (2015) is the only author who proposes a model that finds a globally optimal policy to the SRCPSP. The reason for this might be that finding such an optimal policy seems to be computationally intractable, and therefore most authors focus on finding optimal or near optimal policies within predefined classes of policies. Different classes of policies have been introduced in the literature such as the class of priority policies, the class of early start policies, the class of (linear) pre-selective policies, the class of activity-based policies and the class of pre-processor policies (Ashtiani et al., 2011; Ballestín and Leus, 2009; Möhring et al., 1984, 1985; Stork, 2001).

The second body of research focuses on proactive and reactive scheduling. Traditionally, a proactive and reactive project scheduling approach is a two stage approach. The first stage is to construct a schedule, which is called the *baseline* schedule, that is as *robust*¹ as possible against a certain type of uncertainty (different types of uncertainty, such as activity duration uncertainty and resource availability uncertainty, have been introduced by several authors in the literature (Demeulemeester and Herroelen, 2011)). A number of justifications for the necessity of a baseline schedule have been discussed in the literature (Herroelen, 2005). The second stage is to reschedule (react) reasonably whenever a *conflict* in the ongoing schedule occurs. A conflict refers to the situation where the schedule is no longer feasible.

Many authors published papers dealing with proactive and/or reactive resource-constrained project scheduling problems in the last two decades. We first cite Leus and Herroelen (2005) who show that scheduling with a single conflict and in the presence of precedence constraints is strongly NP-hard even in a single machine environment. The same authors, in a different study (Herroelen and Leus, 2004), propose exact methods (MIP formulations) to construct solution robust baseline schedules for the case in which only one conflict occurs. We also cite Van de Vonder et al. (2005) who seek to understand the trade-off between quality and solution robustness, Van de Vonder et al. (2006) who propose a *resource flow-dependent float factor* (RFDFF) heuristic to construct solution robust schedules and Van de Vonder et al. (2008) who develop new heuristic procedures such as the *starting time criticality* (STC) heuristic to solve the proactive RCPSP. Finally, we cite Lamas and Demeulemeester (2015) who introduce a MIP formulation for the chance-constrained RCPSP (C-C RCPSP) which is the chance-constrained version of the sequence-based RCPSP formulation introduced by Alvarez-Valdes and Tamarit (1993).

Besides the two aforementioned strands of research, the bi-criterion RCPSP under uncertainty also has been studied in the last decade. The most common bi-criterion objective function is the combination of the minimization of the makespan and the optimization of a predictive robustness measure. Al-Fawzan and Haouari (2005) propose a tabu search algorithm for the bi-criterion RCPSP where the robustness measure is the total *free slack*². In a different approach, Abbasi et al. (2006) present a simulated annealing method that minimizes a linear function of the makespan and the total free slack. Moreover, Chtourou and Haouari (2008) define twelve different robustness measures and propose a two-stage robust schedule generation approach. We refer to Demeulemeester and Herroelen (2011) for detailed and broad classifications of different methods dealing with RCPSP under uncertainty.

Despite the popularity of the traditional proactive and reactive procedures, we noticed that in

¹Robustness refers to the ability of tolerating variabilities that may affect the feasibility of a schedule.

²The free slack is the amount of time that each activity can be delayed without delaying any other activity.

almost all of the studies, the authors forgot two important aspects. First, these authors ignored the impact of the choice of the reactive scheduling policy on the optimality of the baseline schedule and vice versa. The only exception is the study by [Deblaere et al. \(2011\)](#) where the proactive and reactive scheduling problems are solved simultaneously using a two-stage approach which outputs a baseline schedule and a corresponding reasonably efficient early-start policy. Not only does the procedure provide a simulation-based heuristic solution, but also the reactive policy is selected from a class of early start policies which extremely limits the flexibility of the procedure. Second, these authors simply assume that the number of reactions does not have any effect on the robustness of the baseline schedule and on that of the reactive policy. In fact, this assumption is inaccurate because of the same reasons that motivate the necessity of the baseline schedule. Moreover, we have to consider that not only each reaction costs some money, but it also damages the business credibility of the contractor.

What we suggest is an alternative method that resembles many robust optimization methods in the literature ([Gabrel et al., 2014](#)), especially *recoverable robust optimization* ([Liebchen et al., 2009](#)) and *adjustable robust optimization* ([Ben-Tal et al., 2003](#); [Shapiro, 2011](#)). We try to tackle a *proactive and reactive resource-constrained project scheduling problem* (PR-RCPSP) considering the aforementioned forgotten aspects in project scheduling. The basic idea behind PR-RCPSP is to select a schedule from a given set of previously generated schedules and then foresee all required transitions to other schedules from the same set for each possible resource or precedence infeasibility³. The combination of a baseline schedule and a set of required transitions is called a *proactive and reactive policy*. The detailed definition of the proactive-and-reactive policy is given in the next section.

In robust scheduling, the idea of generating multiple schedules is not new. For example, the *contingent scheduling approach*, which has been classified as a proactive scheduling approach ([Chaari et al., 2014](#); [Herroelen and Leus, 2005](#)), is based on the generation of multiple baseline schedules. Another example is the very recent research by [Akkan et al. \(2016\)](#) in which a pool of solutions is generated and different neighborhoods of solutions are evaluated.

With respect to the current literature, the novelty of our contribution is three-fold. First, we introduce a novel way of reacting to conflicts. Second, we model proactive and reactive project scheduling as a single problem where the objective function includes the cost of a baseline schedule as well as the costs of reactions and is to find an optimal proactive-and-reactive policy. Third, we introduce four dynamic programming approaches (Models 1-4) that can solve the problem over different subsets of policies. Note that Model 1, which is outperformed by Models 2-4, is an intermediate model that simplifies the understanding of Models 2-4.

2 Definition and problem statement

In this section, we introduce the proactive and reactive resource-constrained project scheduling problem. The following data are given for each instance of the problem:

- A set $N = \{0, 1, \dots, n+1\}$ of activities where activities 0 and $n+1$ are the dummy start and dummy end activities. Each activity $i \in N \setminus \{0, n+1\}$ has an independent stochastic integer duration, $p_i^{\min} \leq \tilde{p}_i \leq p_i^{\max}$, which follows a discrete distribution. Notice that $\tilde{p}_0 = \tilde{p}_{n+1} = 0$. We assume that the realized duration of the activity is only known when the execution of the activity is completed. The vector $\tilde{\mathbf{p}} = (\tilde{p}_0, \tilde{p}_1, \dots, \tilde{p}_{n+1})$ can be represented by a finite supporting set $\mathfrak{P} = \{\mathbf{p}_1, \dots, \mathbf{p}_{|\mathfrak{P}|}\}$ of realizations where each realization $\mathbf{p}_l$ is represented by

³A schedule S is called infeasible if at least one activity i cannot be started at S_i without violating any resource or precedence constraint.

a vector $\mathbf{p}_l = (p_{l0}, p_{l1}, \dots, p_{l(n+1)}) \in \mathfrak{P}$ of realized durations. The probability of occurrence of the realization $\mathbf{p}_l$ is shown by $\pi(\tilde{\mathbf{p}} = \mathbf{p}_l)$. Beware that $\mathfrak{P}$ can be a very large set, and as such is only used in this section to facilitate introducing a compact formulation. Instead, our solving approaches directly use the information provided by the given discrete distributions.

- A set $\mathcal{R}$ of resource constraints. Each job i requires r_{ik} units of resource type $k \in \mathcal{R}$ during its processing time. The resource availability of resource type k is denoted by R_k .
- A set $E = \{(i, j) | i, j \in N\}$ of precedence constraints among activities. The pair $(i, j) \in E$ indicates that activity j cannot be started before activity i is completed.

2.1 Solution representation

A single schedule S , which is a vector of starting times, cannot represent a solution because activity durations are stochastic. Instead, solutions are *proactive-and-reactive policies* (PR-policies). We introduce two different descriptions of a PR-policy in [Section 2.1.1](#) and [Section 2.1.2](#).

2.1.1 Rule-based description

A PR-policy Π can be described by a set of decisions rules that dictate certain transitions among schedules. This set of decision rules that describe PR-policy Π is referred to as the *rule-based description* of PR-policy Π . At each decision moment, some information becomes known, *i.e.* the set of completed activities, the set of ongoing activities, the starting times of the ongoing and completed activities and the current execution time. This information defines the *state of the execution*. Each decision in PR-policy Π is associated with a certain state of the execution. In the following, we discuss the relation between a PR-policy and a traditional proactive and reactive scheduling solution.

In most reactive scheduling methods in the literature, a *reaction* is a set of rules that dictate certain rescheduling of activities in order to resolve conflicts. Alternatively, we consider a *reaction* as the transition from one schedule to another schedule. Let $U(S, t)$ represent the set of not yet started activities in schedule S at time t . A transition from schedule S to schedule S' can be done if and only if $U(S, t) = U(S', t)$ and $S_i = S'_i$ for all $i \in N \setminus U(S, t)$.

A PR-policy Π not only determines reactions, but also selects one schedule as the baseline schedule. The baseline schedule, shown by $S^{[0]\Pi}$, can be seen as the result of the reaction from the start of the execution where no information is available, which is by definition infeasible to any given realization.

Let $\mathcal{S}(\mathbf{p}_l)$ be the set of all feasible schedules for a given realization $\mathbf{p}_l$. To address all possible schedules, we introduce the set $\mathcal{S}$ of schedules that can be constructed as follows $\mathcal{S} = \mathcal{S}(\mathbf{p}_1) \cup \dots \cup \mathcal{S}(\mathbf{p}_{|\mathfrak{P}|})$. Since in the general case, $\mathcal{S}$ is a very large set, we define a much smaller set $\mathbf{S} \subset \mathcal{S}$ as the set of schedules. For practical reasons, we use the set $\mathbf{S}$ rather than $\mathcal{S}$ in our models (Models 1-4) in [Section 3](#). Although the choice of the set $\mathbf{S}$ influences the performance of Models 1-4 a lot, it is not the focus of this paper to find the best set of schedules. However, in [Section 3.1.5](#), we will explain a few heuristic approaches that construct reasonably good sets of schedules.

2.1.2 Chain-based description

In order to start the execution of the project, a PR-policy Π selects a baseline schedule ($S^{[0]\Pi}$). The execution is continued until the first conflict occurs. Since it is not known which realization will occur, PR-policy Π uses only the information that is available in the current state of the execution to transit to a second schedule. After that, the execution is continued until another conflict occurs.

Then, PR-policy Π enforces another transition to a third schedule. PR-policy Π makes a series of transitions until either all conflicts are resolved or there is no schedule to which a transition from the current schedule is possible. The latter situation is referred to as a *deadend* and is denoted by Ω . Notice that deadends only occur when we have a limited number of schedules in our set of schedules.

For each realization $\mathbf{p}_l$ (in case it occurs), PR-policy Π enforces a chain (series) of $\nu_{\Pi,l}$ reactions. This chain is denoted by $\Phi_{\Pi,l}$:

$$\Phi_{\Pi,l} : S^{[0]_{\Pi}} \xrightarrow{t=t_1} S^{[1]_{\Pi,l}} \xrightarrow{t=t_2} S^{[2]_{\Pi,l}} \xrightarrow{t=t_3} \dots \xrightarrow{t=t_{\nu_{\Pi,l}}} S^{[\nu_{\Pi,l}]_{\Pi,l}}$$

The occurred realization and its associated chain of reactions is known only when the execution of the project has ended (either all activities are completed or a deadend is reached). Obviously, the baseline schedule must be the same for all chains (realizations).

A chain with a deadend is called a *deadchain*. We introduce the parameter $\gamma_{\Pi,l}$ which equals one if $\Phi_{\Pi,l}$ is a deadchain and equals zero otherwise. We compute the incurred combined cost of PR-policy Π for each chain $\Phi_{\Pi,l}$. This combined cost, denoted by $f(\Pi, l)$, is also used as a measure for stability and robustness in [Section 4](#):

$$f(\Pi, l) = w_b \times S_{n+1}^{[0]_{\Pi,l}} + \sum_{k=1}^{\nu_{\Pi,l}} \left(\sum_{i \in U(S^{[k-1]_{\Pi,l}}, t_k)} w_{ik} |S_i^{[k]_{\Pi,l}} - S_i^{[k-1]_{\Pi,l}}| + w_r \right) + \gamma_{\Pi,l} M$$

where $w_b \geq 0$ is the cost per unit time for the completion time of the baseline schedule, $w_r \geq 0$ is the fixed cost incurred with each reaction, $w_{ik} \geq 0$ is the stability cost of activity i in the k th reaction and M (an arbitrary large value) represents the penalty of having a deadend. This cost function, consists of three parts: the cost of the baseline schedule, the cost of a series of reactions and the cost of having a deadend.

A PR-policy Π , which is described by a set of decision rules, can also be described by its associated set of chains:

$$\Pi : \{\Phi_{\Pi,1}, \Phi_{\Pi,2}, \dots, \Phi_{\Pi,|\mathfrak{P}|}\}$$

This set of chains, which describes Π , is referred to as the *chain-based description* of Π . This description enables us to compute the expected combined cost of PR-policy Π :

$$\sum_{l=1}^{|\mathfrak{P}|} \pi(\tilde{\mathbf{p}} = \mathbf{p}_l) f(\Pi, l)$$

Notice that the occurring realization is not known until the end of the execution and therefore the chain-based description, unlike its rule-based counterpart, is too confusing to be used in practice. However, it can be perfectly used in our problem formulation.

A PR-policy Π *includes* a schedule if that schedule has been part of at least one of the chains that describe Π . It might be interesting to know what is the worst case size of a PR-policy, *i.e.* the number of schedules that are included in that PR-policy. Let $m_{\max} = \max_{i \in N} \{m_i\}$ and m_i is the number of modes (*i.e.*, the number of different durations) of activity i .

Theorem 1. *The size of the chain-based description of a PR-policy is bounded by $nm_{\max}|\mathfrak{P}|$.*

Proof: Each PR-policy includes $|\mathfrak{P}|$ chains. This number, though, can be very large. For each chain (or realization), we can have at most $nm_{\max}$ reactions. Therefore, we infer that the maximum number of schedules that are included in a PR-policy is bounded by $nm_{\max}|\mathfrak{P}|$. $\square$

2.2 Conceptual formulation

We define Π as the set of all possible PR-policies (*i.e.*, the set of PR-policies that can be constructed by $\mathcal{S}$). Our problem (PR-RCPSP) which is denoted for simplicity by P is formulated as follows:

$$P : \min_{\Pi \in \Pi} \sum_{l=1}^{|\mathcal{P}|} \pi(\tilde{\mathbf{p}} = \mathbf{p}_l) f(\Pi, l)$$

Theorem 2. P is an NP-hard problem in the strong sense.

Proof: We prove the NP-hardness of our problems by reduction from the deterministic RCPSP. Consider an instance of the deterministic RCPSP with activity duration vector $\mathbf{p}^{[det]}$. Set $w_b = 1$, $\pi(\tilde{p}_i = p) = 1$ if $p = p_i^{[det]}$ and $\pi(\tilde{p}_i = p) = 0$ if $p \neq p_i^{[det]}$. Since the durations are deterministic, no reaction will be in the optimal policy for this instance. Therefore, P finds a baseline schedule with the minimum makespan. $\square$

P is still a very difficult problem to solve. We propose to solve P_1 that searches over a smaller set of policies:

$$P_1 : \min_{\Pi \in \Pi_1} \sum_{l=1}^{|\mathcal{P}|} \pi(\tilde{\mathbf{p}} = \mathbf{p}_l) f(\Pi, l)$$

In the above formulation, Π_1 is the set of all PR-policies that only include the schedules in $\mathbf{S}$. Model 1 which is proposed in [Section 3.1](#) optimally solves P_1 in a reasonable computation time where $|\mathbf{S}| \leq 2000$. Readers might question why we limit ourselves to PR-policies which select schedules only from $\mathbf{S}$. We are completely aware that selecting schedules only from $\mathbf{S}$ may lead to inflexible reactions and thus high combined costs, especially when $|\mathbf{S}|$ is too small. This confusion vanishes when we introduce problems P_2 , P_3 and P_4 in [Sections 3.2-3.4](#) that search over much larger classes of PR-policies and are optimally solved by Models 2, 3 and 4, respectively.

2.3 Example project

Let us describe an *example project* that provides a better understanding of our problem. Note that this example project will be also used and extended in other sections of this paper. Our example project includes 8 non-dummy activities and one resource type with an availability of 8. [Figure 1](#) depicts the precedence relations and the resource requirements of the activities of the project. Each node represents an activity, each arc represents a precedence relation and the number above each node shows the resource requirement for that activity.

The activity durations for this example follow the distributions given in [Table 1](#). Each fractional value represents the probability that $\tilde{p}_i$ equals a certain value p . For instance, the probability that $\tilde{p}_1 = 1$ is 0.4. The probabilities in each row must obviously sum up to 1. [Table 1](#) also include the stability cost of each activity. For this example, it is assumed that $w_{i,k} = w_{i,0}$ for all $k = 0, 1, \dots, nm_{\max}$. The scheduling parameters w_b , w_r and M are given as follows: $w_b = 40$, $w_r = 20$ and $M = 1000$.

In total, there are $2^5 \times 3^3 = 864$ realizations. Each realization occurs with a certain computable probability of occurrence. For example, $\mathbf{p}_1 = (0, 2, 8, 3, 5, 7, 5, 4, 2, 0)$ is a vector of durations which represents a single realization and its chance of occurrence is approximately 0.054% (the corresponding probabilities are indicated in bold in [Table 1](#)).

The set of schedules ($\mathbf{S}$) which is given in [Table 2](#) contains only 10 schedules. Consider a PR-policy Π_1 with the following rule-based description:

Figure 1: The precedence network for the example project

	$\pi(\tilde{p}_i = p)$									$w_{i,1}$
	$p = 0$	$p = 1$	$p = 2$	$p = 3$	$p = 4$	$p = 5$	$p = 6$	$p = 7$	$p = 8$	
$\tilde{p}_0$	1	-	-	-	-	-	-	-	-	-
$\tilde{p}_1$	-	0.4	0.4	0.2	-	-	-	-	-	4
$\tilde{p}_2$	-	-	-	-	-	-	0.3	0.5	0.2	4
$\tilde{p}_3$	-	-	-	0.6	0.4	-	-	-	-	7
$\tilde{p}_4$	-	-	-	0.1	0.5	0.4	-	-	-	1
$\tilde{p}_5$	-	-	-	-	-	-	-	0.2	0.8	4
$\tilde{p}_6$	-	-	-	-	-	0.4	0.6	-	-	1
$\tilde{p}_7$	-	-	-	0.5	0.5	-	-	-	-	1
$\tilde{p}_8$	-	-	0.7	0.3	-	-	-	-	-	1
$\tilde{p}_9$	1	-	-	-	-	-	-	-	-	38

Table 1: The distribution of activity durations and the weights of the activities for the first reaction

- The baseline schedule is S^9 .
- At time 2 if the current schedule is S^9 and activities 1 and 4 are ongoing, then react to schedule S^8 ; otherwise continue S^9 .
- At time 4 if the current schedule is S^8 and activities 3 and 4 are ongoing, then react to schedule S^5 ; otherwise continue S^8 .

While the rule-based description is suitable for managerial purposes, a chain-based description is needed to compute the cost of the PR-policy. The chain-based description of PR-policy Π_1 is given as follows:

$$\Pi_1 : \begin{cases} \Phi_{\Pi_1, k_1} : S^9 & \forall \mathbf{p}_{k_1} \in \mathfrak{P}_1 \\ \Phi_{\Pi_1, k_2} : S^9 \xrightarrow{t=2} S^8 \rightarrow \Omega & \forall \mathbf{p}_{k_2} \in \mathfrak{P}_2 \\ \Phi_{\Pi_1, k_3} : S^9 \xrightarrow{t=2} S^8 & \forall \mathbf{p}_{k_3} \in \mathfrak{P}_3 \\ \Phi_{\Pi_1, k_4} : S^9 \xrightarrow{t=2} S^8 \xrightarrow{t=4} S^5 \rightarrow \Omega & \forall \mathbf{p}_{k_4} \in \mathfrak{P}_4 \\ \Phi_{\Pi_1, k_5} : S^9 \xrightarrow{t=2} S^8 \xrightarrow{t=4} S^5 & \forall \mathbf{p}_{k_5} \in \mathfrak{P}_5 \end{cases}$$

	S^k									
	S^1	S^2	S^3	S^4	S^5	S^6	S^7	S^8	S^9	S^{10}
S_0^k	0	0	0	0	0	0	0	0	0	0
S_1^k	0	0	0	0	0	0	0	0	0	0
S_2^k	1	1	0	1	5	0	7	4	2	7
S_3^k	3	3	4	4	3	3	3	3	5	5
S_4^k	0	0	4	0	0	7	0	0	0	9
S_5^k	6	6	7	7	7	7	7	7	9	14
S_6^k	6	6	7	7	7	12	5	7	9	14
S_7^k	7	8	7	8	12	12	14	12	11	15
S_8^k	11	13	13	12	15	15	17	15	15	20
S_9^k	13	15	15	15	17	18	19	18	18	23

Table 2: A given set $\mathbf{S}$ for the example project

where

$$\begin{aligned}
\mathfrak{P}_1 &= \{\mathbf{p}_k | p_{k,1} \leq 2, k = 1, \dots, |\mathfrak{P}|\}, \\
\mathfrak{P}_2 &= \{\mathbf{p}_k | p_{k,1} > 2, p_{k,4} \leq 4, S^8 \text{ is infeasible for } \mathbf{p}_k, k = 1, \dots, |\mathfrak{P}|\}, \\
\mathfrak{P}_3 &= \{\mathbf{p}_k | p_{k,1} > 2, p_{k,4} \leq 4, S^8 \text{ is feasible for } \mathbf{p}_k, k = 1, \dots, |\mathfrak{P}|\}, \\
\mathfrak{P}_4 &= \{\mathbf{p}_k | p_{k,1} > 2, p_{k,4} > 4, S^5 \text{ is infeasible for } \mathbf{p}_k, k = 1, \dots, |\mathfrak{P}|\}, \\
\mathfrak{P}_5 &= \{\mathbf{p}_k | p_{k,1} > 2, p_{k,4} > 4, S^5 \text{ is feasible for } \mathbf{p}_k, k = 1, \dots, |\mathfrak{P}|\},
\end{aligned}$$

and $\mathfrak{P}_1 \cup \mathfrak{P}_2 \cup \mathfrak{P}_3 \cup \mathfrak{P}_4 \cup \mathfrak{P}_5 = \mathfrak{P}$. Note that the total number of chains is 864. Each chain Φ_{Π_1, k_x} in the above description represents a set of identical chains. For example, Φ_{Π_1, k_1} represents a set of 576 chains associated with all realizations $\mathbf{p}_k \in \mathfrak{P}_1$ ($|\mathfrak{P}_1| = 576$).

PR-policy Π_1 selects S^9 as the baseline schedule. If $\mathbf{p}_1$ occurs, S^9 never becomes infeasible and thus no reaction is needed. However, if $\mathbf{p}_2 = (0, 3, 8, 3, 4, 8, 5, 4, 2, 0)$ occurs, S^9 becomes infeasible at time 2 (Figure 2a) and Π_1 dictates a transition from S^9 to S^8 to resolve the infeasibility. The resource profile of the two schedules is depicted in Figure 2. The cost of this reaction is computed as follows:

$$\begin{aligned}
\sum_{i \in U(S^9, 2)} w_{i,1} |S_i^8 - S_i^9| + w_r &= |4 - 2| \times 4 + |3 - 5| \times 7 + |7 - 9| \times 4 + |7 - 9| \times 1 + |12 - 11| \times 1 \\
&+ |15 - 15| \times 1 + |18 - 18| \times 38 + 20 = 53.
\end{aligned}$$

If $\mathbf{p}_3 = (0, 3, 8, 3, 5, 8, 5, 4, 2, 0)$ occurs, Π_1 dictates a reaction from S^9 to S^8 to resolve the infeasibility at time 2 (see Figure 2a-2b for the associated resource profiles) and a reaction from S^8 to S^5 to resolve the infeasibility at time 4 (see Figure 2c-2d for the associated resource profiles). The cost of the second reaction is computed as follows:

$$\sum_{i \in U(S^8, 4)} w_{i,2} |S_i^5 - S_i^8| + w_r = 62.$$

For each realization $\mathbf{p}_l$ in $\mathfrak{P}_1$, we compute the cost of the associated chain as follows: $f(\Pi_1, l) = 40 \times 18 = 720$. The cumulative probability of occurrence of realizations in $\mathfrak{P}_1$ equals $\sum_{l \in \mathfrak{P}_1} \pi(\tilde{\mathbf{p}} = \mathbf{p}_l) = 0.80$. Similarly we compute:

- For each $\mathbf{p}_l \in \mathfrak{P}_2$, $f(\Pi_1, l) = 720 + 53 + 1000 = 1773$
- For each $\mathbf{p}_l \in \mathfrak{P}_3$, $f(\Pi_1, l) = 720 + 53 = 773$
- For each $\mathbf{p}_l \in \mathfrak{P}_4$, $f(\Pi_1, l) = 720 + 53 + 62 + 1000 = 1835$
- For each $\mathbf{p}_l \in \mathfrak{P}_5$, $f(\Pi_1, l) = 720 + 53 + 62 = 835$

Also, $\sum_{l \in \mathfrak{P}_2} \pi(\tilde{\mathbf{p}} = \mathbf{p}_l) = 0.06$, $\sum_{l \in \mathfrak{P}_3} \pi(\tilde{\mathbf{p}} = \mathbf{p}_l) = 0.06$, $\sum_{l \in \mathfrak{P}_4} \pi(\tilde{\mathbf{p}} = \mathbf{p}_l) = 0.0576$ and $\sum_{l \in \mathfrak{P}_5} \pi(\tilde{\mathbf{p}} = \mathbf{p}_l) = 0.0224$. Therefore, the expected combined cost of PR-policy Π_1 is computed as follows:

$$\sum_{l=1}^{|\mathfrak{P}|} \pi(\tilde{\mathbf{p}} = \mathbf{p}_l) f(\Pi_1, l) = 0.80 \times 720 + 0.06 \times (1773 + 773) + 0.0576 \times 1835 + 0.0224 \times 835 = 853.16$$

It is worth mentioning that PR-policy Π_1 happens to be the optimal PR-policy to P_1 for this example.

3 Solution methodology

Problem P can be modeled as a *Markov decision process* (MDP). We propose four MDPs in this section, namely Models 1-4. Model 1, which will be explained in [Section 3.1](#), optimally solves P_1 for a given set of schedules. Nevertheless, this model only searches over a limited set of PR-policies. Therefore, we also provide Models 2-4 in the following subsections, which search over much larger sets of PR-policies than that of Model 1. While Model 2 is very similar to Model 1 and Model 4 is very similar to Model 3, the second pair have fundamental differences with the first one. These fundamental differences enable Model 3 and Model 4 to output deadend-free PR-policies.

This is not the first time MDPs are used to solve stochasticity in project scheduling. We refer interested readers to [Creemers et al. \(2010\)](#) and [Creemers \(2015\)](#).

3.1 Model 1

As mentioned above, Model 1 is an MDP that optimally solves P_1 . Our model description is structured as follows. First, the state representation is given. Then, the transitions among states are introduced. Next, the recursion system is described. After that, we introduce a graph representation by means of an example and finally, an algorithm is proposed to generate the set of schedules.

3.1.1 State representation

The combination (S, t, O, ν) represents a state in Model 1 where S represents the current schedule, t is the current time, O denotes the set of ongoing activities at time t and ν is the total number of reactions that previously occurred upon entrance into the state.

States are labeled either as feasible or infeasible. Let $J(S, t)$ be the set of all activities in S that are supposed to be started at time t .

Definition (Feasible vs infeasible states). A state (S, t, O, ν) is *feasible* if it is possible to execute all activities in $J(S, t) \cup O$ in parallel and is *infeasible* otherwise.

[Figure 2a](#) depicts the resource profile of S^9 at time 2 for realization $\mathbf{p}_2$. The associated state is $(S^9, 2, \{1, 4\}, 0)$. This state is an infeasible state since the set of activities $J(S^9, 2) \cup \{1, 4\} = \{1, 2, 4\}$ cannot be executed in parallel. [Figure 2b](#) depicts the resource profile of S^8 at time 2 for realization $\mathbf{p}_2$. The associated state is $(S^8, 2, \{1, 4\}, 0)$ which is a feasible state because the set of activities $J(S^8, 2) \cup \{1, 4\} = \{1, 4\}$ can be executed in parallel.

(a) Resource profile for schedule S^9 which becomes infeasible at time 2 for realizations $\mathbf{p}_2$ and $\mathbf{p}_3$

(b) Resource profile for schedule S^8 which is feasible at time 2 for realizations $\mathbf{p}_2$ and $\mathbf{p}_3$

(c) Resource profile for schedule S^8 which becomes infeasible at time 4 for realization $\mathbf{p}_3$

(d) Resource profile for schedule S^5 which is feasible at time 4 for realization $\mathbf{p}_3$

Figure 2: An example reaction. The infeasibility is resolved by a transition from schedule S^9 to schedule S^8 at time 2. The length of each box indicates the minimum possible duration for the associated activity. Other possible durations of an activity are indicated by vertical bars together with their probabilities of occurrence. Note that at time t , the lengths of the associated boxes of finished activities represent their realized duration. Also notice that the minimum length of the box associated with an ongoing activity cannot be less than or equal to its realized duration so far.

3.1.2 Transitions

Upon leaving a state, we enter another state. Entering state (S, t, O, ν) means that schedule S is considered for execution, the current time is t , the activities in O are ongoing and we have already reacted to conflicts ν times. Depending on whether we face a conflict (the current state is infeasible) or not (the current state is feasible), and also depending on the realization and the PR-policy we choose, we could enter a different state.

A transition between two states is possible if and only if an arc exists between those two states. The states are connected via two types of arcs: *chance* arcs and *decision* arcs. Chance arcs leave feasible states whereas decision arcs leave infeasible states.

If state (S, t, O, ν) is feasible, it means no conflict is happening and there is no need for any reaction at that decision moment. In this case, a transition is required to a new state (S, t', O', ν) where t' is the next decision moment in S after t and $O' \subseteq O \cup J(S, t)$. The set O' might differ depending on which realization occurs. Therefore, the number of chance arcs leaving a state might be more than one. Upon leaving a feasible state, we may enter one of the states to which a chance arc exists from the left state. A chance arc that connects (S, t, O, ν) to (S, t', O', ν) is denoted by $(S, t \rightarrow t', O \rightarrow O', \nu)$. While transiting from (S, t, O, ν) to (S, t', O', ν) , exactly one of the following options happens to each activity $i \in O \cup J(S, t)$:

- activity i that was ongoing at time t is still ongoing at time t' , and therefore $(i \in O \cap O')$,
- activity i that was ongoing at time t finishes before time t' , and therefore $(i \in O \setminus O')$,
- activity i that is started at time t is still ongoing at time t' , and therefore $(i \in J(S, t) \cap O')$,
- activity i that is started at time t finishes before time t' , and therefore $(i \in J(S, t) \setminus O')$,

Because the distribution of each activity is known, we can compute the probability of transition over a certain chance arc $\pi(S, t \rightarrow t', O \rightarrow O', \nu)$ as follows:

$$\begin{aligned} \pi(S, t \rightarrow t', O \rightarrow O', \nu) &= \prod_{i \in O \cap O'} \frac{\pi(S_i + \tilde{p}_i > t')}{\pi(S_i + \tilde{p}_i > t)} \times \prod_{i \in J(S, t) \cap O'} \pi(t + \tilde{p}_i > t') \\ &\times \prod_{i \in O \setminus O'} \left(1 - \frac{\pi(S_i + \tilde{p}_i > t')}{\pi(S_i + \tilde{p}_i > t)}\right) \times \prod_{i \in J(S, t) \setminus O'} (1 - \pi(t + \tilde{p}_i > t')) \end{aligned}$$

Note that the summation of the probabilities of chance arcs leaving a feasible state must equal 1. If $O = \emptyset$ and $J(S, t) = \{n + 1\}$, then the execution of the project has already completed and no chance arc leaves (S, t, O, ν) . In this case, (S, t, O, ν) is an *endstate*.

Figure 2b depicts the resource profile associated with $(S^8, 2, \{1, 4\}, 0)$. If we move to the next decision moment, which is time 3, depending on which realization occurs, we arrive to one of two different states: $(S^8, 2, \{4\}, 0)$ or $(S^8, 2, \emptyset, 0)$. Figure 3 demonstrates the chance arcs leaving state $(S^8, 2, \{1, 4\}, 0)$. We compute the probability of each chance arc leaving $(S^8, 2, \{1, 4\}, 0)$ as follows:

$$\begin{aligned} \pi(S^8, 2 \rightarrow 3, \{1, 4\} \rightarrow \{4\}, 0) &= \frac{0.90}{1} \times \left(1 - \frac{0}{0.2}\right) = 0.90 \\ \pi(S^8, 2 \rightarrow 3, \{1, 4\} \rightarrow \emptyset, 0) &= \left(1 - \frac{0}{0.2}\right) \times \left(1 - \frac{0.90}{1}\right) = 0.10 \end{aligned}$$

Upon leaving an infeasible state, we decide to transit to a feasible state. However, not all such transitions are valid. There does exist a decision arc from an infeasible state (S, t, O, ν) to a feasible

Figure 3: Chance arcs leaving $(S^8, 2, \{1, 4\}, 0)$

Figure 4: Transition from S to S' at time t

state $(S', t, O, \nu + 1)$ if and only if $U(S, t) = U(S', t)$ and $S_i = S'_i$ for all $i \in N \setminus U(S, t)$. [Figure 4](#) depicts a Gantt chart illustration of a valid transition from (S, t, O, ν) to $(S', t, O, \nu + 1)$. In this Gantt chart representation of the schedules, F represents the finished activities, O represents the ongoing activities at time t and $U(S, t)$ represents the activities in the schedule S that have not started their execution before time t . Note that although it might not be clear in the Gantt charts, the starting times of activities in F and O should be exactly the same for the two schedules. In reality, a transition from S to S' at time t means that the management team decides to continue the execution of the project according to schedule S' (instead of the infeasible schedule S) from time t until a further conflict occurs. These transitions are called *normal transitions*. The cost of a transition from (S, t, O, ν) to $(S', t + c, O, \nu + 1)$ equals:

$$w_r + \sum_{i \in U(S, t)} (w_{i(\nu+1)} |S'_i - S_i|).$$

In the following subsections, we propose other types (namely, flexible and cut transitions) of reactions.

For each infeasible state (S, t, O, ν) , we introduce $\Gamma_1(S, t, O)$ that represents the set of all schedules to which a valid transition from (S, t, O, ν) exists. In other words, if $S' \in \Gamma_1(S, t, O)$, then there exists a transition from (S, t, O, ν) to $(S', t, O, \nu + 1)$. The larger the set $\Gamma_1(S, t, O)$, the more flexibility we have in reacting. We noticed that if $\mathbf{S}$ is too small, the set $\Gamma_1(S, t, O)$ might be an empty set for some combinations of S , t and O . If a state (S, t, O, ν) is infeasible and $\Gamma_1(S, t, O) = \emptyset$, then we have no reaction possibility. Such a state is called a *deadstate*. Notice that a deadstate is the last state in a deadchain and its corresponding schedule is a deadend.

Consider the infeasible state $(S^9, 2, \{1, 4\}, 0)$. We compute the set of schedules that are valid candidates for reaction: $\Gamma_1(S^9, 2, \{1, 4\}) = \{S^5, S^7, S^8\}$. [Figure 5](#) demonstrates the decision arcs leaving the infeasible state $(S^9, 2, \{1, 4\}, 0)$. The number depicted over each arc represents the cost of the associated reaction. For example, the cost of reacting to schedule S^5 is 95.

Figure 5: Decision arcs leaving $(S^9, 2, \{1, 4\}, 0)$

3.1.3 Markov recursion

We introduce $d_1(S \rightarrow S', t, O, \nu \rightarrow \nu + 1)$ as the expected cost until the end of execution if we decide to transit from (S, t, O, ν) to $(S', t, O, \nu + 1)$ in Model 1. We also introduce $c_1(S, t, O, \nu)$ as the expected cost until the end of execution upon leaving each feasible state (S, t, O, ν) . First, we compute the expected cost until the end of the execution for each decision arc:

$$d_1(S \rightarrow S', t, O, \nu \rightarrow \nu + 1) = w_r + \sum_{i \in U(S, t)} (w_{i(\nu+1)} |S'_i - S_i|) + c_1(S', t, O, \nu + 1) \quad (1)$$

Then, the best decision and its associated expected cost are computed as follows:

$$S^* = \arg \min_{S' \in \Gamma_1(S, t, O)} \{d_1(S \rightarrow S', t, O, \nu \rightarrow \nu + 1)\} \quad (2)$$

$$d_1^*(S, t, O, \nu) = d_1(S \rightarrow S^*, t, O, \nu \rightarrow \nu + 1) \quad (3)$$

Let $\mathcal{F}_1$ be the set of all feasible states and $\mathcal{I}_1$ be the set of all infeasible states in Model 1. Likewise, the set of all endstates and the set of all deadstates are denoted by $\mathcal{E}_1$ and $\mathcal{D}_1$, respectively. The function $c_1(S, t, O, \nu)$ is computed as follows (remember that the expected cost of an endstate is zero):

$$\begin{aligned} c_1(S, t, O, \nu) = & \sum_{(S, t', O', \nu) \in \mathcal{F}_1 \setminus \mathcal{E}_1} \pi(S, t \rightarrow t', O \rightarrow O', \nu) * c_1(S, t', O', \nu) \\ & + \sum_{(S, t', O', \nu) \in \mathcal{I}_1 \setminus \mathcal{D}_1} \pi(S, t \rightarrow t', O \rightarrow O', \nu) * d_1^*(S, t', O', \nu) \\ & + \sum_{(S, t', O', \nu) \in \mathcal{D}_1} \pi(S, t \rightarrow t', O \rightarrow O', \nu) * M \end{aligned} \quad (4)$$

Note that in the above equation, t' is always the next decision moment after t and O' is whatever set that satisfies $O' \subseteq O \cup J(S, t)$.

We need to select one of the schedules in $\mathbf{S}$ as the baseline schedule. To that end, we can use the information provided while solving our model. The cost of choosing schedule S as the baseline schedule equals $c_1(S, 0, \emptyset, 0) + w_b S_{n+1}$. Therefore, we select the baseline schedule (S^{base}) as follows:

$$S^{\text{base}} = \arg \min_{S \in \mathbf{S}} \{c_1(S, 0, \emptyset, 0) + w_b S_{n+1}\} \quad (5)$$

It is straightforward to see that the recursion system (1)-(5) optimally solves P_1 and the optimal objective value equals $c_1(S^{\text{base}}, 0, \emptyset, 0) + w_b S_{n+1}^{\text{base}}$. Let $O_{\max}$ be the set of ongoing activities with maximum cardinality. The total number of states is in the order of $O(n^2 2^{|O_{\max}|} |\mathbf{S}|)$ and the maximum number of arcs (chance arcs or decision arcs) leaving a state equals $\max\{2^{|O_{\max}|}, |\mathbf{S}|\}$. Therefore, Model 1 can be solved in $O(n^2 2^{|O_{\max}|} |\mathbf{S}| \max\{2^{|O_{\max}|}, |\mathbf{S}|\})$ time.

to the infeasible state $(S^1, 1, \{1, 4\}, 0)$, where activities 1 and 4 are ongoing at decision moment 1 and activity 2 is about to start, or a transition to the feasible state $(S^1, 1, \{4\}, 0)$, where activity 4 is ongoing at decision moment 1 and activity 2 is about to start. The former transition occurs with a probability of 60% while the latter one occurs with a probability of 40%. Knowing the expected costs until the end of execution upon leaving the states $(S^1, 1, \{1, 4\}, 0)$ and $(S^1, 1, \{4\}, 0)$, we can simply compute the expected cost until the end of execution upon leaving the state $(S^1, 0, \emptyset, 0)$ as follows:

$$\begin{aligned} c_1(S^1, 0, \emptyset, 0) &= 0.40 \times c_1(S^1, 1, \{4\}, 0) + 0.60 \times d_1^*(S^1, 1, \{1, 4\}, 0) = 0.40 \times 985.15 + 0.60 \times 472.93 \\ &= 677.82 \end{aligned}$$

Figure 6 also indicates a part of an optimal PR-policy Π_1 in dark black. PR-policy Π_1 , as stated in the definition of PR-policy, enforces certain decisions for each infeasible state. For example, among all arcs leaving the start node, Π_1 chooses $(S^9, 0, \emptyset, 0)$ and therefore $S^{[0]n_1} = S^9$. Also upon leaving $(S^9, 2, \{1, 4\}, 0)$, which is an infeasible state and represents a conflict in schedule S^9 at time 2 when activities $\{1, 4\}$ are ongoing and activity 2 is about to start, Π_1 dictates a transition to $(S^8, 2, \{1, 4\}, 1)$. Notice that the expected cost until the end of execution upon leaving the state $(S^9, 2, \{1, 4\}, 0)$ is computed as follows:

$$d_1^*(S^9, 2, \{1, 4\}, 0) = \min\{612.80 + 53, 720 + 111, 720 + 95\} = 665.80$$

3.1.5 The set of schedules

In Algorithm 1, we present a *pool generation procedure* (PGP) which outputs a set of schedules given a set of initial schedules. In this procedure, max_S is the exact number of schedules that will be generated. Due to the high computational complexity of our methods, we limit ourselves to $max_S \leq 2000$. A number of subprocedures are also used in PGP which will be explained in the following lines: $rndSchedule(S)$ returns one schedule which is selected randomly from the set S of all already generated schedules, $rndRealization(\mathfrak{P})$ returns a random realization, $infeas(S, \mathbf{p}_l, t)$ returns *true* if schedule S becomes infeasible at decision moment t for realization $\mathbf{p}_l$ and returns *false* otherwise and $nextDM(S, t)$ returns the next decision moment after t in schedule S . The subprocedure $react(S, \mathbf{p}_l, t)$ which reacts to the conflict at decision moment t is a *robust parallel schedule generation scheme* (RP-SGS) where the duration vector is $\hat{\mathbf{p}}$ and the scheme's priority rule is the *earliest baseline activity starting time* (EBST).

Depending on what initial set we select, what value we choose for max_S and what is the reaction policy, PGP results in different sets of schedules. The initial set could be one of the following: 1) a set including one single schedule that is the optimal schedule for the deterministic case where the duration vector is $\hat{\mathbf{p}}$ (we refer to this set as DET). 2) a set including 13 schedules that are the outcome of the *starting time criticality* (STC) procedure proposed by Van de Vonder et al. (2008), each with a different α selected from $\{1, 1.025, 1.050, \dots, 1.25, 1.275, 1.3\}$ (we refer to this set as STC). 3) A set of fifty schedules generated by Algorithm 2. In this algorithm, the subprocedure $DH(\cdot)$ optimally solves the deterministic RCPSP using the branch-and-bound method proposed by Demeulemeester and Herroelen (1992, 1997) (we refer to this set as SMP which stands for sampling).

3.2 Model 2

In Section 3.1, Model 1 which optimally solves P_1 was proposed. It is not difficult to see that the optimal PR-policy to P_1 has a reasonable combined cost only if the given set of schedules is very

Algorithm 1 Pool generation procedure (PGP)

Input: A set $\mathbf{S}^{\text{init}}$ of initial schedules

```
1:  $\mathbf{S} = \mathbf{S}^{\text{init}}$ 
2:  $n_{\mathbf{S}} = |\mathbf{S}^{\text{init}}|$ 
3: while  $n_{\mathbf{S}} < \max_{\mathbf{S}}$  do
4:    $S = \text{rndSchedule}(\mathbf{S})$ 
5:    $\mathbf{p}_l = \text{rndRealization}(\mathfrak{P})$ 
6:    $t = 0$ 
7:   while  $t \leq S_{n+1}$  do
8:     if  $\text{infeas}(S, \mathbf{p}_l, t)$  then
9:        $S' = \text{react}(S, \mathbf{p}_l, t)$ 
10:       $\mathbf{S} = \mathbf{S} \cup \{S'\}$ 
11:       $S = S'$ 
12:       $n_{\mathbf{S}} = n_{\mathbf{S}} + 1$ 
13:      if  $n_{\mathbf{S}} = \max_{\mathbf{S}}$  then
14:        break
15:      end if
16:    end if
17:     $t = \text{nextDM}(S, t)$ 
18:  end while
19: end while
```

Output: $\mathbf{S}$

Algorithm 2 An initial pool generation scheme

```
1:  $n_{\mathbf{S}} = 0$ 
2: while  $n_{\mathbf{S}} < 50$  do
3:   generate 4 random realization  $\mathbf{p}_1, \dots, \mathbf{p}_4$ 
4:    $\mathbf{p}_{\max} = \max\{\mathbf{p}_1, \dots, \mathbf{p}_4\}$ 
5:    $S = DH(\mathbf{p}_{\max})$ 
6:    $\mathbf{S}^{\text{init}} \leftarrow S$ 
7:    $n_{\mathbf{S}} = n_{\mathbf{S}} + 1$ 
8: end while
```

Output: $\mathbf{S}^{\text{init}}$

large. However, as we have already mentioned, Model 1 is not computationally tractable when the given set of schedules is large. Therefore, we only can solve instances of P_1 where the given set of schedules is relatively small, which results in a high probability of having deadends in its optimal PR-policy and a very high optimal combined cost. This motivates us to investigate what would be an alternative model that searches over a larger set of PR-policies and yet is computationally tractable.

The poor performance of Model 1 is mainly because of its inflexibility in reactions. In Model 2, we try to relax some criteria to allow more flexibility in reactions. The inflexibility in normal transitions comes from the fact that the starting times of all started activities should be exactly the same in both schedules. Upon leaving the infeasible state (S, t, O, ν) , often no schedule $S^\#$ such that $(S \rightarrow S^\#, t, O, \nu \rightarrow \nu + 1)$ is a normal transition exists in $\mathbf{S}$. However, we noticed that the states of the execution and the costs until the end of the execution for the two states $(S^\#, t, O, \nu + 1)$

Figure 7: Transition from S to S'' at time t

and $(S'', t+c, O, \nu+1)$ are exactly the same if $S_i^\# = S_i'' + c, \forall i \in O$ and $U(S^\#, t) = U(S'', t+c)$. Therefore, if such S'' exists in the set $\mathbf{S}$, then upon leaving the infeasible state (S, t, O, ν) , one may transit from (S, t, O, ν) to $(S'', t+c, O, \nu+1)$ to resolve the infeasibility. This transition is called a *flexible transition*.

In the new model, a transition from (S, t, O, ν) to $(S'', t+c, O, \nu+1)$ is possible if $S_i = S_i'' + c, \forall i \in O$ and $U(S, t) = U(S'', t+c)$ and either of the two criteria below is true:

- $t+c$ is a decision moment in S'' and $(S'', t+c, O, \nu+1)$ is feasible.
- $t+c$ is not a decision moment in S'' .

Note that c is whatever constant value (including 0). We introduce $\Gamma_2(S, t, O)$ as the set of all pairs (S'', c) for which a transition from (S, t, O, ν) to $(S'', t+c, O, \nu+1)$ is possible. Figure 7 illustrates the condition in which a transition from S to S'' is valid. In reality, a flexible transition results in a new schedule $(S^\#)$ which might not exist within the set of schedules, but can be implied by the schedules in the set. In other words, the flexible transition from (S, t, O, ν) to $(S'', t+c, O, \nu+1)$ means that the managerial team decides to continue the execution of the project according to $S^\#$ (the schedule at the very bottom of Figure 7), from time t until another conflict occurs in the schedule. The cost of a transition from (S, t, O, ν) to $(S'', t+c, O, \nu+1)$ equals:

$$w_r + \sum_{i \in U(S, t)} (w_{i(\nu+1)} |S_i'' + c - S_i|).$$

The recursion system for Model 2 can be computed in the same fashion as that of Model 1 except that, in Model 2, we use $\Gamma_2(S, t, O)$ instead of $\Gamma_1(S, t, O)$ which is used in Model 1. This recursion system is given in Appendix A.1. Model 2 is very similar to Model 1 with only a difference in the valid reactions.

Let us define a new set $\mathbf{\Pi}_2$ as the set of all PR-policies that can be constructed using the schedules in S and allowing flexible transitions. Note that $\mathbf{\Pi}_1 \subseteq \mathbf{\Pi}_2$. We introduce problem P_2 as follows:

(b) Model 2: a flexible transition to $(S^9, 8, \{2, 3\}, 1)$ resolves the infeasibility in state $(S^4, 7, \{2, 3\}, 0)$

Figure 8: The difference in reaction possibilities between Model 1 and Model 2

$$P_2 : \min_{\Pi \in \Pi_2} \sum_{l=1}^{|\mathbb{P}|} \pi(\tilde{\mathbf{p}} = \mathbf{p}_l) f(\Pi, l)$$

Model 2 (the recursion system (15)-(19)) optimally solves P_2 . Since $\Pi_1 \subseteq \Pi_2$ Model 2 results in a solution that is at least as good as the solution provided by Model 1 if started from the same set of schedules. Although Model 2 searches over a much larger set of PR-policies than that in Model 1, the two models share the same worst-case computational complexity. Nevertheless, Model 2 requires more computational resources (computational steps and memory requirements) than Model 1.

3.2.1 Graph representation of flexible transition

Figure 8 depicts the difference in the reaction possibilities of the two models. Model 1 provides no reacting transition to resolve the infeasibility in state $(S^4, 7, \{2, 3\}, 0)$ (Figure 8a) whereas Model 2 provides a flexible transition (Figure 8b). In the flexible transition between $(S^4, 7, \{2, 3\}, 0)$ and $(S^9, 8, \{2, 3\}, 1)$, the starting times of the ongoing activities (activity 2 and activity 3) in S^4 are $S_2^4 = 1$ and $S_3^4 = 4$ and the starting time of the ongoing activities in S^9 are $S_2^9 = 2$ and $S_3^9 = 5$. If we choose $c = 1$, then the first condition is met. $U(S^4, 7) = U(S^9, 8)$ which satisfies the second condition and $7 + c = 7 + 1 = 8$ is not a decision moment in S^9 which fulfills the last condition. Thus this flexible transition is valid.

3.2.2 Enhancement by introducing base states

We introduce *base states* as follows. A base state which is denoted by (S, t, O) includes all "child" states (S, t, O, ν) with $\nu \geq 0$. All states within a base state are called siblings. The decision arcs and the chance arcs that are leaving a state are very similar to those of its siblings. For instance, consider the two states (S, t, O, ν) and $(S, t, O, \nu + 1)$. Assume that the two chance arcs $(S, t \rightarrow t', O \rightarrow O', \nu)$ with probability of occurrence 0.4 and $(S, t \rightarrow t', O \rightarrow O'', \nu)$ with probability of occurrence 0.6 are leaving the state (S, t, O, ν) . Likewise, the two chance arcs $(S, t \rightarrow t', O \rightarrow O', \nu + 1)$ with probability of occurrence 0.4 and $(S, t \rightarrow t', O \rightarrow O'', \nu + 1)$ with probability of occurrence 0.6 are leaving the state $(S, t, O, \nu + 1)$. Also, duplicated decision arcs leave infeasible siblings. In order to reduce computational resources needed to solve the problem, we only compute and store chance arcs and/or decision arcs for the base state and use them for its child states.

Moreover, we can enhance the algorithm even further for the special case where $w_{ik} = w_{i,k+1}$, $\forall i \in N, k \geq 0$. If so, then the cost of each state within a certain base state equals the cost of any

of its siblings. In other words, we have

$$c_2(S, t, O, 0) = c_2(S, t, O, 1) = c_2(S, t, O, 2) = \dots \quad (6)$$

Once a cost of a state is known, we can use its cost for all of its siblings. Therefore, the algorithm requires much less computational resources.

3.3 Model 3

Even though Model 2 is meant to reduce the number of deadstates, computational results show that there are still reasonable amounts of deadstates in the resulting network of Model 2, which raises serious questions on the effectiveness of Model 2. In this section, we introduce Model 3 whose network is proven to be deadstate-free.

3.3.1 State representation

We introduce a new state representation for Model 3: states are defined by (cu, co, de, ν) . A *cu* (an abbreviation of *a project cut*) indicates the execution situation of the project at a certain decision moment and is defined by the trio $(F, O, \boldsymbol{\rho})$ where F denotes the set of finished activities, O represents the set of ongoing activities and $\boldsymbol{\rho}$ represents the vector of *elapsed times* of the ongoing activities. The elapsed time of an activity is the time passed since the starting of that activity. A *co* (an abbreviation of *a continuation*) indicates the starting times of the remainder of the project at a certain decision moment and is defined by the pair $(I, \boldsymbol{\rho})$ where I denotes the set of not yet started activities and $\boldsymbol{\rho}$ represents the relative starting times of the not yet started activities. If the starting time of an activity i is S_i and the continuation starts at time t , then the relative starting time of activity i for this continuation will be $S_i - t$. The value $de \geq 0$ determines the delay between the moment of the cut and the start of the continuation. Each state (S, t, O, ν) in Model 1 (or Model 2) can be linearly mapped to its corresponding state (cu, co, de, ν) in Model 3. Figure 9 shows how to convert state $(S^7, 7, \{6\}, 0)$ of Model 1 to its corresponding state $(cu_{(S^7, 7, \{6\})}, co_{(S^7, 7)}, 0, 0)$ of Model 3 where $cu_{(S^7, 7, \{6\})} = (\{1, 3, 4\}, \{6\}, (*, *, *, *, *, *, 2, *, *, *))$ and $co_{(S^7, 7)} = (\{2, 5, 7, 8, 9\}, (*, *, 0, *, *, 0, *, 7, 10, 12))$. Readers should beware that, in our examples, instead of using numerical indexes for cuts and continuations (e.g. cu_1 and co_1), we use parametric combination (S, t, O) as the index of a cut (e.g. $cu_{(S^7, 7, \{6\})}$) and parametric combination (S, t) as the index of a continuation (e.g. $co_{(S^7, 7)}$). This parametric indexing approach provides much more information to the reader than a numerical indexing approach. However, readers must note that a single unique cut (or a single unique continuation) can be represented by different parametric combinations. For instance, the cut $(\{0\}, \{1, 4\}, (*, 1, *, *, 1, *, *, *, *, *))$ can be represented by both $cu_{(S^1, 1, \{1, 4\})}$ and $cu_{(S^2, 1, \{1, 4\})}$.

3.3.2 Transitions

Let $J(co, t)$ be the set of activities that start t time units after the start of the continuation ($J(co, t) = \{\rho_i = t | i \in I\}$). A state (cu, co, de, ν) is feasible if either $de > 0$ or scheduling $J(co, 0)$ at the moment of cu , while the activities in O are ongoing, is resource and precedence feasible. Remind that, if $de > 0$ then there is a positive gap between the moment of the cut and the start of the continuation and thus it is guaranteed that no activity is started until the next decision moment.

If (cu, co, de, ν) is a feasible state, then a set of chance arcs leave this state. On one hand, if $de > 0$, then there exist chance arcs $(cu \rightarrow cu', co, de \rightarrow 0, \nu)$ with $cu = (F, O, \boldsymbol{\rho})$ and $cu' = (F', O', \boldsymbol{\rho}')$ such that

Figure 9: An example of a cut and a continuation

- $F \subseteq F'$, $O' \subseteq O$ and $F \cup O = F' \cup O'$,
- if $i \in O \setminus O'$, then $de + \varrho_i \geq p_i^{\min}$,
- if $i \in O \cap O'$, then $\varrho'_i = de + \varrho_i \leq p_i^{\max}$.

The probability of occurrence of each chance arc $(cu \rightarrow cu', co, de \rightarrow 0, \nu)$ is computed as follows:

$$\pi(cu \rightarrow cu', co, de \rightarrow 0, \nu) = \prod_{i \in O \cap O'} \frac{\pi(\tilde{p}_i > de + \varrho_i)}{\pi(\tilde{p}_i > \varrho_i)} \times \prod_{i \in O \setminus O'} \left(1 - \frac{\pi(\tilde{p}_i > de + \varrho_i)}{\pi(\tilde{p}_i > \varrho_i)}\right).$$

Consider $(cu_{(S^7, 6, \{6\})}, co_{(S^7, 7)}, 1, 0)$. This state is a feasible state since $de = 1$. There is only one chance arc leaving this state which is $(cu_{(S^7, 6, \{6\})} \rightarrow cu_{(S^7, 7, \{6\})}, co_{(S^7, 7)}, 1 \rightarrow 0, \nu)$. The probability of occurrence of this chance arc is computed as follows (note that $O \setminus O' = \{6\} \setminus \{6\} = \emptyset$):

$$\pi(cu_{(S^7, 6, \{6\})} \rightarrow cu_{(S^7, 7, \{6\})}, co_{(S^7, 7)}, 1 \rightarrow 0, \nu) = \frac{\pi(\tilde{p}_6 > 1 + 1)}{\pi(\tilde{p}_6 > 1)} = 1.$$

On the other hand, if $de = 0$ in a feasible state (cu, co, de, ν) , then there exists a set of chance arcs $(cu \rightarrow cu', co \rightarrow co', 0, \nu)$ with $cu = (F, O, \varrho)$, $cu' = (F', O', \varrho')$, $co = (I, \rho)$ and $co' = (I', \rho') = co \triangleleft t_{next}$ such that

- $F \subseteq F'$ and $O' \subseteq O \cup J(co, 0)$,
- if $i \in O \setminus O'$, then $t_{next} + \varrho_i \geq p_i^{\min}$,
- if $i \in O' \cap O$, then $\varrho'_i = t_{next} + \varrho_i \leq p_i^{\max}$
- if $i \in J(co, 0) \cap O'$, then $\varrho'_i = t_{next} \leq p_i^{\max}$
- if $i \in J(co, 0) \setminus O'$, then $t_{next} \geq p_i^{\min}$

where $t_{next} = \min\{\rho_i > 0 | i \in I\}$ and $\triangleleft$ is a left-shift operator. This operator left-shifts each activity in a continuation by a constant value and removes any activity that cannot be left-shifted. The probability of occurrence of the chance arc $(cu \rightarrow cu', co \rightarrow co', 0, \nu)$ is computed as follows:

$$\begin{aligned} \pi(cu \rightarrow cu', co \rightarrow co', 0, \nu) &= \prod_{i \in O \cap O'} \frac{\pi(\tilde{p}_i > t_{next} + \varrho_i)}{\pi(\tilde{p}_i > \varrho_i)} \times \prod_{i \in J(co, 0) \cap O'} \pi(\tilde{p}_i > t_{next}) \\ &\times \prod_{i \in O \setminus O'} \left(1 - \frac{\pi(\tilde{p}_i > t_{next} + \varrho_i)}{\pi(\tilde{p}_i > \varrho_i)}\right) \times \prod_{i \in J(co, 0) \setminus O'} (1 - \pi(\tilde{p}_i > t_{next})) \end{aligned}$$

An example might help to clarify clarification of the above formulations. Consider state $(cu_{(S^7,7,\{6\})}, co_{(S^7,7)}, 0, 0)$ (see Figure 9). It is resource and precedence feasible to schedule the activities in $J(co_{(S^7,7)}, 0) \cup \{6\} = \{2, 5, 6\}$ and thus the state is feasible. In S^7 , the next decision moment after 7 is 14. Since the schedule is feasible and no reaction is needed, the continuation at the next decision moment will be $co_{(S^7,7)} \triangleleft 7 = co_{(S^7,14)}$ (in general, $co_{(S,t)} \triangleleft x = co_{(S,t+x)}$ always holds). The detailed computations of this left shift are given below:

$$\begin{aligned} co_{(S^7,7)} \triangleleft 7 &= (\{2, 5, 7, 8, 9\}, (*, *, 0, *, *, 0, *, 7, 10, 12)) \triangleleft 7 = (\{7, 8, 9\}, (*, *, *, *, *, *, *, 0, 3, 5)) \\ &= co_{(S^7,14)} \end{aligned}$$

In this example, activities 2 and 7 are removed because they cannot be left-shifted. The following four chance arcs leave state $(cu_{(S^7,7,\{6\})}, co_{(S^7,7)}, 0, 0)$:

- $(cu_{(S^7,7,\{6\})} \rightarrow cu_{(S^7,14,\emptyset)}, co_{(S^7,7)} \rightarrow co_{(S^7,14)}, 0, 0)$ with a probability of occurrence of 0.16,
- $(cu_{(S^7,7,\{6\})} \rightarrow cu_{(S^7,14,\{2\})}, co_{(S^7,7)} \rightarrow co_{(S^7,14)}, 0, 0)$ with a probability of occurrence of 0.04,
- $(cu_{(S^7,7,\{6\})} \rightarrow cu_{(S^7,14,\{5\})}, co_{(S^7,7)} \rightarrow co_{(S^7,14)}, 0, 0)$ with a probability of occurrence of 0.64,
- $(cu_{(S^7,7,\{6\})} \rightarrow cu_{(S^7,14,\{2,5\})}, co_{(S^7,7)} \rightarrow co_{(S^7,14)}, 0, 0)$ with a probability of occurrence of 0.16.

A transition (decision arc) between $(cu, co, 0, \nu)$ and $(cu, co', de', \nu+1)$ that is shown by $(cu, co \rightarrow co', 0 \rightarrow de', \nu \rightarrow \nu+1)$ is valid only if the former state is infeasible and the later one is feasible. The cost of this transition, which is referred to as a *cut transition*, equals:

$$w_r + \sum_{i \in I} (w_{i(\nu+1)} |\rho_i - (de' + \rho'_i)|).$$

The set of all cut transitions also includes all normal and flexible transitions. For each infeasible state $(cu, co, 0, \nu)$ (remind that any state (cu, co, de, ν) with $de \geq 1$ is certainly feasible), we introduce $\Gamma_3(cu, co)$ that represents the set of all pairs (co', de') for which a cut transition from $(cu, co, 0, \nu)$ to $(cu, co', de', \nu+1)$ exists. Note that from each infeasible state $(cu, co, 0, \nu)$, there is at least one valid cut transition to state $(cu, co, 1, \nu+1)$, and therefore it is guaranteed that the network in Model 3 is deadstate-free. Some experiments show that such a network might explode. Therefore, we enforce $de' \leq 1$ in our experiments.

Consider state $(S^7, 14, \{2, 5\}, 2)$ in Model 2 and its associated state $(cu_{(S^7,14,\{2,5\})}, co_{(S^7,14)}, 0, 2)$ in Model 3. Figure 10a depicts state $(S^7, 14, \{2, 5\}, 2)$ which is a deadstate in Model 2. While Model 2 fails to resolve the infeasibility in state $(S^7, 14, \{2, 5\}, 2)$, Model 3 proposes six different possible cut transitions to resolve the infeasibility in $(cu_{(S^7,14,\{2,5\})}, co_{(S^7,14)}, 0, 2)$ (see Figure 10b). Notice that three of these transitions (indicated by dashed lines) are simply too inefficient and therefore are dominated by the dominance rule that will be introduced in Section 3.3.4.

3.3.3 Markov recursion

First, we compute the cost until the end of the execution for each decision arc:

$$\begin{aligned} d_3(cu, co \rightarrow co', 0 \rightarrow de', \nu \rightarrow \nu+1) &= w_r + \\ &\sum_{i \in I} (w_{i(\nu+1)} |\rho_i - (de' + \rho'_i)|) + c_3(cu, co', de', \nu+1) \end{aligned} \quad (7)$$

Then, the cost associated to the best decision is computed as follows:

$$(co^*, de^*) = \arg \min_{(co', de') \in \Gamma_3(cu, co)} \{d_3(cu, co \rightarrow co', 0 \rightarrow de', \nu \rightarrow \nu+1)\} \quad (8)$$

$$d_3^*(cu, co, 0, \nu) = d_3(cu, co \rightarrow co^*, 0 \rightarrow de^*, \nu \rightarrow \nu+1) \quad (9)$$

1000
 $S^7, 14, \{2, 5\}, 2$

(a) Model 2: no reaction possibility to resolve the infeasibility in state $(S^7, 14, \{2, 5\}, 2)$

(b) Model 2: cut transitions that resolve the infeasibility in state $(cu_{(S^7, 14, \{2, 5\})}, co_{(S^7, 14)}, 0, 2)$

Figure 10: The difference in reaction possibilities between Model 2 and Model 3

We introduce the set $\mathcal{F}_3$, $\mathcal{I}_3$ and $\mathcal{E}_3$ as the set of all feasible states, the set of all infeasible states and the set of all endstates in Model 3, respectively. The function $c_3(cu, co, de, \nu)$ is computed as follows:

$$\begin{aligned}
 c_3(cu, co, de, \nu) = & \sum_{(cu', co', de', \nu) \in \mathcal{F}_3 \setminus \mathcal{E}_3} \pi(cu \rightarrow cu', co \rightarrow co', de \rightarrow de', \nu) * c_3(cu', co', de', \nu) \\
 & + \sum_{(cu', co', 0, \nu) \in \mathcal{I}_3} \pi(cu \rightarrow cu', co \rightarrow co', de \rightarrow 0, \nu) * d_3^*(cu', co', 0, \nu)
 \end{aligned} \tag{10}$$

Similarly to the previous models, we need to select one of the schedules in $\mathbf{S}$ as the baseline schedule. Lets introduce $cu^0 = (\emptyset, \emptyset, ())$. We select the baseline schedule (S^{base}) as follows:

$$S^{\text{base}} = \arg \min_{S \in \mathbf{S}} \{c_3(cu^0, co_{(S, 0)}, 0, 0) + w_b S_{n+1}\} \tag{11}$$

Let us define the set $\mathbf{\Pi}_3$ as the set of all PR-policies that can be constructed using the schedules in $\mathbf{S}$ and allowing cut transitions. Note that $\mathbf{\Pi}_1 \subseteq \mathbf{\Pi}_2 \subseteq \mathbf{\Pi}_3$. We introduce problem P_3 as follows:

$$P_3 : \min_{\Pi \in \mathbf{\Pi}_3} \sum_{l=1}^{|\mathbb{P}|} \pi(\tilde{\mathbf{p}} = \mathbf{p}_l) f(\Pi, l)$$

The recursion system (7)-(11) optimally solves P_3 and the optimal objective value equals $c_3(cu^0, co_{(S^{\text{base}}, 0)}, 0, 0) + w_b S_{n+1}^{\text{base}}$. Because $\mathbf{\Pi}_1 \subseteq \mathbf{\Pi}_2 \subseteq \mathbf{\Pi}_3$, Model 3 provides a solution which is at least as good as the solution provided by Model 2 or Model 1 for the same set of starting schedules.

In Model 3, we need to generate all possible continuations before starting the recursion. In order to avoid duplicates, for each continuation we make sure that the smallest relative starting time of an activity is zero, otherwise, we left-shift that continuation until this condition is fulfilled. For example, if $co = (\{1, 2, 3\}, (2, 5, 7))$, then we left-shift co by two units ($co = co \triangleleft 2$) and therefore we will have $co = (\{1, 2, 3\}, (0, 3, 5))$.

3.3.4 Improvement by removing dominated decision arcs

In Model 3, many transition cuts are possible. Thus, plenty of decision arcs are leaving each state. However, many of these decision arcs are very costly to choose and hence will never be part of an optimal solution. The following dominance rule removes those decision arcs that are too costly and can be dominated by at least one other decision arc.

Dominance rule 1. *Assume that one has partially computed the cost of an infeasible state $(cu, co, 0, \nu)$. That means he/she has computed the expected costs until the end if he/she chooses some (but not all) of the decision arcs. Let $(\hat{co}, \hat{de})$ be the best decision so far for this infeasible state. For each remaining decision (co', de') , if*

$$w_r + \sum_{i \in I} (w_{i(\nu+1)} |\rho_i - (de' + \rho'_i)|) \geq d_3(cu, co \rightarrow \hat{co}, 0 \rightarrow \hat{de}, \nu \rightarrow \nu + 1),$$

then (co', de') is certainly not the best decision for (cu, co, de, ν) and its corresponding arc $(cu, co \rightarrow co', 0 \rightarrow de', \nu \rightarrow \nu + 1)$ can be eliminated.

To support the above dominance rule, it suffices to remind that the following inequality holds:

$$d_3(cu, co \rightarrow co', 0 \rightarrow de', \nu \rightarrow \nu + 1) \geq w_r + \sum_{i \in I} (w_{i(\nu+1)} |\rho_i - (de' + \rho'_i)|)$$

Consider the decision arcs in [Figure 10b](#). The cost until the end of the execution for decision arc $(cu_{(S^7, 14, \{2, 5\})}, co_{(S^7, 14)} \rightarrow co_{(S^1, 7)}, 0 \rightarrow 1, 2 \rightarrow 3)$ is $99 + 17.40 = 116.40$ which is already lower than the cost of the transition from $(cu_{(S^7, 14, \{2, 5\})}, co_{(S^7, 14)}, 0, 2)$ to $(cu_{(S^7, 14, \{2, 5\})}, co_{(S^2, 8)}, 1, 3)$. Thus, decision arc $(cu_{(S^7, 14, \{2, 5\})}, co_{(S^7, 14)} \rightarrow co_{(S^2, 8)}, 0 \rightarrow 1, 2 \rightarrow 3)$ is dominated.

3.4 Model 4

As we already mentioned in [Section 3.3](#), many very costly decision arcs exist in Model 3 and only some of them can be dominated by [Dominance rule 1](#). We introduce Model 4 which is very similar to Model 3 with the only difference that the decisions are generated in a much smarter fashion.

For each combination of $(cu, co, 0)$, which represents an infeasible schedule, we compute a number of continuations using RP-SGS (note that preliminary results suggest that RP-SGS generally outperforms *robust serial SGS* (RS-SGS)) with different lists (EBST and *lowest activity number* (LAN)) and different sets of durations ($\mathbf{p}^{50\%}$, $\mathbf{p}^{60\%}$, $\mathbf{p}^{70\%}$ and $\mathbf{p}^{80\%}$) where $\mathbf{p}^{x\%}$ is computed as follows:

$$\begin{aligned} \mathbf{p}^{x\%} &= (p_0^{x\%}, p_1^{x\%}, \dots, p_{n+1}^{x\%}) \\ p_i^{x\%} &= \min\{p_i | p_i^{\min} \leq p_i \leq p_i^{\max}, \pi(\tilde{p}_i \leq p_i) \geq x\%\} \end{aligned}$$

For each different choice of a reaction policy, a set of durations and a list, we possibly end up in a different continuation. In order to avoid duplications we ensure that the smallest relative starting time of an activity is zero, otherwise we left-shift that continuation until this condition

is fulfilled. For each combination of (cu, co) , the above computations provide the set of possible reactions $(\Gamma_4(cu, co))$ with at most 8 different pairs (co', de') . Each pair (co', de') represents a *wise* transition. In comparison with $\Gamma_3(cu, co)$ (which is the set of possible reactions in Model 3), this set includes a much smaller number of pairs. However, the quality of the pairs in $\Gamma_4(cu, co)$ is, on average, meant to be much better than that of those in $\Gamma_3(cu, co)$.

The recursion system of Model 4 can be computed in the same fashion as that of Model 3 except that in Model 4 we use $\Gamma_4(cu, co)$ instead of $\Gamma_3(cu, co)$. This recursion system is explained in more detail in [Appendix A.2](#).

Let us define a new set Π_4 as the set of all PR-policies that can be constructed using the baseline schedules in S and allowing wise transitions. We introduce problem P_4 as follows:

$$P_4 : \min_{\Pi \in \Pi_4} \sum_{l=1}^{|\mathbb{P}|} \pi(\tilde{\mathbf{p}} = \mathbf{p}_l) f(\Pi, l)$$

Model 4 (the recursion system (20)-(24)) optimally solves P_4 .

4 Computational results

All models (Models 1-4) have been implemented in Visual C++ 2010. All computational results, unless stated otherwise, were obtained on a laptop Dell Latitude with 2.6 GHz Core(TM) i7-3720QM processor, 8GB of RAM and running under Windows 10.

4.1 Instance generation

Our models are tested on a set of 48 instances that are composed of the PSPLIB instances. Only instances with 30 non-dummy activities are considered in this experiment. PSPLIB is a class of instances for the deterministic RCPSP ([Kolisch and Sprecher, 1997](#)), thus they need to be modified to suit our problem. The following modifications are applied on this set of instances: the activity durations $\tilde{p}_i$ for each non-dummy activity i follow a discretized beta distribution with shape parameters 2 and 5 that is mapped over the interval $[0.75\hat{p}_i, 1.625\hat{p}_i]$ where $\hat{p}_i$ is the duration of activity i that is given in the original instance ([Van de Vonder et al., 2007](#)).

The base activity weights w_{j0} for $j \in \{1, 2, \dots, n\}$ are obtained from a discrete triangular distribution with $Pr(w_j = q) = (21 - 2q)\%$ for $q \in \{1, 2, \dots, 10\}$ ([Van de Vonder et al., 2007](#)). This distribution results in a higher probability for low weights. The average weight w_{avg} then equals 3.85 which is used to calculate the weight of the dummy end activity $w_{n+1,0} = 38 \approx 10w_{\text{avg}}$. The non-base activity weights are computed as follows: $w_{jk} = \lambda^k w_{j0}; k = 1, \dots, 30$. The value of λ is chosen from $\{1, 1.1, 1.2\}$.

In order to reduce the number of experiments, we only consider the instances from the set J30 of PSPLIB with the following filename syntax: J30x.1 ($x = 1, \dots, 48$).

4.2 Measures of stability and robustness

In this section, we introduce and discuss two measures that are used as comparison measures:

- Combined cost (CC): the expected proactive and reactive cost for each PR-policy Π which is computed as follows:

$$\sum_{l=1}^{|\mathbb{P}|} \pi(\tilde{\mathbf{p}} = \mathbf{p}_l) f(\Pi, l)$$

- Recovery robustness (RR): the probability of not having a deadend which is computed by:

$$\sum_{l=1}^{|\mathfrak{P}|} \pi(\tilde{\mathbf{p}} = \mathbf{p}_l)(1 - \gamma_{\Pi, l})$$

4.3 Results for our proposed models

In this section, we discuss the results of our four proposed models. For each model, we report the average CPU time (in seconds) required to solve the problem (CPU), the average combined cost (CC), the average recovery robustness (RR) and the average number of states (#states). For Model 1 and Model 2 we also report the average number of chance arcs (#c arcs) and the average number of decision arcs (#d arcs), whereas for Model 3 and Model 3 we report the average number of cuts (#cuts) and the average number of continuations (#cons).

4.3.1 Model 1

We run Model 1 on the set of instances explained above and over the nine different sets of schedules. These nine different sets are constructed using three different initial sets of schedules and three different values for $|\mathbf{S}|$. For this experiment, we choose $w_b = 50$, $\lambda = 1$, $w_r = 100$ and $w_{de} = 1000$.

Table 3 depicts an overview of the results obtained for Model 1 on the nine different sets of schedules. Note that in order to reduce the space required to store all arcs, states with the same S , t and O are grouped into bigger states (called base state) and then arcs are only added among base states. Therefore, it might be possible that the number of states is greater than the number of chance arcs and/or the number of decision arcs.

There are improving trends in both combined cost and recovery robustness by increasing the size of the set of schedules. The only exception is when we generate schedules using DET. In this case, the combined cost is fluctuating while the recovery robustness is improved by increasing the size of the set of schedules. This behavior is considered to be strange because more schedules results in more reaction possibilities and smaller numbers of deadends and thus it is expected that the combined cost is decreased. This strange behavior is possible when the expected cost imposed by the added reaction possibilities (when we use the larger set of schedules) is larger than the expected cost imposed by the deadends (when we use the smaller set of schedules). For example, when we use DET to generate 500 schedules, recovery robustness is around 0.07. When we use DET to generate 1000 schedules, recovery robustness is around 0.09 which means that the probability of having a deadend is around 2% less than when we only generate 500 schedules. This means that in 2% of cases, instead of paying w_{de} which is the cost of the deadends, we must pay the costs of reactions. If the expected cost of these reactions can be higher than the expected cost of deadends (*i.e.* $1000 \times 0.02 = 20$), then this strange behavior might occur.

Also, we notice that both combined cost and recovery robustness are significantly improved, while CPU time increases, by using STC or SMP instead of DET as the initial set of schedules. On average, it takes around 20 seconds to generate 13 schedules in the set STC and around 4.5 seconds to generate 50 schedules in SMP. Finally, we observe that as we increase the number of schedules, the number of states and the number of chance arcs are linearly increased whereas the increase in the number of decision arcs is rather quadratic.

4.3.2 Model 2

The summary of the results obtained for Model 2 are reported in Table 4. For this experiment, similarly to Model 1, we choose $w_b = 50$, $\lambda = 1$, $w_r = 100$ and $w_{de} = 1000$. Obviously, Model 2

$\mathcal{S}^{init}$	$ \mathcal{S} $	<i>CPU</i>	<i>CC</i>	<i>RR</i>	<i>#states</i>	<i>#c arcs</i>	<i>#d arcs</i>
DET	500	1.06	4034.94	0.07	236561	105149	260149
	1000	2.97	4040.95	0.09	519700	217163	1027909
	2000	9.24	4038.69	0.15	1139451	448983	4073017
STC	500	20.96	3946.84	0.42	217502	104098	219300
	1000	24.17	3934.67	0.48	485080	213790	882217
	2000	29.37	3919.82	0.53	1064437	440083	3563275
SMP	500	5.69	4011.25	0.38	273222	106733	223245
	1000	7.09	3988.70	0.41	595881	220948	886837
	2000	13.38	3957.46	0.47	1296020	456841	3533487

Table 3: Summary of the results for Model 1

$\mathcal{S}^{init}$	$ \mathcal{S} $	<i>CPU</i>	<i>CC</i>	<i>RR</i>	<i>#states</i>	<i>#c arcs</i>	<i>#d arcs</i>
DET	500	0.87	3739.34	0.80	61347	135441	1123350
	1000	2.38	3686.56	0.88	126631	284889	4606344
	2000	8.76	3645.05	0.91	260473	596044	18841063
STC	500	20.52	3696.72	0.86	64640	143110	1058972
	1000	22.05	3641.85	0.91	133056	299560	4317168
	2000	27.19	3607.48	0.94	273195	625034	17593573
SMP	500	5.42	3696.85	0.89	71156	167705	987363
	1000	6.67	3649.07	0.92	146259	349139	3974585
	2000	11.39	3614.29	0.94	300199	722934	15943784

Table 4: Summary of the results for Model 2

outperforms Model 1 in both combined cost and recovery robustness. Although Model 2 runs faster than Model 1 when $\lambda = 1$ which is mainly a result of the enhancement techniques explained in [Section 3.2.2](#), it needs more computational resources (both computation steps and memory requirements) than those needed for Model 1 to solve the problem. Similarly to Model 1, increasing the size of the set of schedules and/or switching from DET to either STC or SMP result in both better combined cost and better recovery robustness. We notice that in spite of a huge improvement in recovery robustness (solutions with on average 80-95% recovery robustness), we fail to reach fully recovery-robust solutions (solutions with 100% recovery robustness).

The numbers of states in Model 2 are much smaller than that of Model 1 because in Model 2 we use base states to reduce the computational resources as explained in [Section 3.2.2](#). However, if $\lambda \neq 1$, then the number of states in Model 2 would have been larger than that in Model 1. Unlike states, the number of arcs in Model 2 is larger than the number of arcs in Model 1. Even though the number of chance arcs hasn't increased much in comparison with Model 1, the increase in the number of decision arcs is quite significant.

4.3.3 Model 3

The summary of the results obtained for Model 3 are reported in [Table 5](#). For this experiment, similarly to Models 1 and 2, we choose $w_b = 50$, $\lambda = 1$, $w_r = 100$ and $w_{de} = 1000$. In Model 3, by definition we must have solutions with full recovery robustness, which is also confirmed in the results. Apart from having fully recoverable robust solutions, Model 3 also provides much less costly solutions as well. The improvement on the combined cost is significant if we choose Model 3 instead of Model 2. But these improvements come at a cost. The computational resources needed to solve Model 3 are much higher than those needed for Model 2. For instance, the number of

$\mathcal{S}^{init}$	$ \mathcal{S} $	<i>CPU</i>	<i>CC</i>	<i>RR</i>	<i>#states</i>	<i>#cuts</i>	<i>#cons</i>
DET	500	177.21	3589.07	1.00	2740331	12337	5278
	1000	736.04	3576.22	1.00	5781750	14886	10040
	2000	-	-	-	-	-	-
STC	500	244.45	3562.03	1.00	3663497	17378	6179
	1000	1029.93	3549.13	1.00	7978517	21270	11643
	2000	-	-	-	-	-	-
SMP	500	180.58	3566.60	1.00	3887518	30545	6016
	1000	790.71	3555.25	1.00	8587042	38190	11483
	2000	-	-	-	-	-	-

Table 5: Summary of the results for Model 3

states required to solve Model 3 over the set of 500 schedules constructed by $\mathcal{S}^{init} = DET$ for the given setting is around 45 times the number of states required to solve Model 2 over the same set of schedules. Also, the CPU time required to solve Model 3 is clearly higher than that required to solve Model 2.

4.3.4 Model 4

The summary of the results obtained for Model 4 are reported in Table 6. For this experiment, similarly to Models 1, 2 and 3, we choose $w_b = 50$, $\lambda = 1$, $w_r = 100$ and $w_{de} = 1000$. Model 4 requires a huge amount of memory because the number of cuts and the number of continuations are much larger than those in Model 3. Therefore, we stop Model 4 if we reach a certain number of states. For example, in this experiment, we stop Model 4 if we reach 40 million states. In our implementation Model 4 first computes $c_4(cu^0, co_{(\mathcal{S}^1, 0)}, 0, 0)$, then $c_4(cu^0, co_{(\mathcal{S}^2, 0)}, 0, 0)$, and continues so until it computes $c_4(cu^0, co_{(\mathcal{S}^{|\mathcal{S}|}, 0)}, 0, 0)$ as the last state. For a certain instance, if before reaching the limit of 40 million states, Model 4 computes $c_4(cu^0, co_{(\mathcal{S}^{|\mathcal{S}|}, 0)}, 0, 0)$, then that instance is considered as *solved*, otherwise Model 4 outputs the best $c_4(cu^0, co_{(\mathcal{S}, 0)}, 0, 0) + w_b S_{n+1}$ among those computed so far.

In Model 4, similarly to Model 3, we aim to find solutions with full recovery robustness, which is also confirmed in the results. Model 4 also provides less costly solutions in comparison with all other models. The improvement on the combined cost comes with a large increase of the number of states, the number of cuts and the number of continuations. For instance, the number of states required to solve Model 4 over the set of 500 schedules is around 5 times the number of states required to solve Model 3 over the same set of schedules. However, as we increase the number of schedules from 500 to 1000, the numbers of states in Model 3 are approximately increased by a factor of more than 2 whereas the numbers of states in Model 4 are approximately increased by a factor of only 1.2. The average CPU time required to solve Model 4 is less than that of Model 3 because there are many more arcs per state in Model 3.

4.4 Comparison with a conventional proactive and reactive method

We compare Models 2, 3 and 4 with a combination of conventional proactive and reactive approaches (the reason that we leave out Model 1 in these comparisons is that it performs much worse than the other two models). As a proactive approach we choose a modified version of the STC method (M-STC) which is introduced in Section 4.4.1 and as a reactive approach we choose RP-SGS where EBST is the scheme’s priority rule. We refer to this combination as CONV.

$\mathbf{S}^{init}$	$ \mathbf{S} $	#solved	CPU	CC	RR	#states	#cuts	#cons
DET	500	43	167.59	3580.87	1.00	10158884	35051	500063
	1000	43	212.46	3572.01	1.00	12566521	35738	597405
	2000	39	268.05	3565.42	1.00	15700267	36080	734225
STC	500	41	190.16	3552.42	1.00	10386517	34090	518785
	1000	41	220.60	3544.55	1.00	12257837	34432	606361
	2000	40	273.30	<u>3539.53</u>	1.00	15149351	34849	743418
SMP	500	36	340.14	3558.44	1.00	19269453	84722	901571
	1000	32	380.09	3554.22	1.00	22111245	85310	1013003
	2000	28	437.62	3549.38	1.00	25398410	85863	1158909

Table 6: Summary of the results for Model 4

4.4.1 Modified-STC (M-STC)

The original STC method introduced by [Van de Vonder et al. \(2008\)](#) produces a reasonably robust baseline schedule whose makespan is not larger than $\delta = \alpha \times \text{makespan}$ where *makespan* is the project completion time of the optimal schedule for the deterministic RCPSP. Given different values for α , STC produces different schedules, some being very tight ($1 \leq \alpha < 1.15$), some being very loose ($\alpha \geq 1.3$) and some being neither tight nor loose ($1.15 \leq \alpha < 1.3$). Depending on what values we choose for w_b , w_r and λ , it might be beneficial to have a loose or tight baseline schedule. It is indeed very difficult to incorporate w_r and λ in the computation of the stability cost exploited in the STC method, but it is possible to incorporate w_b in the computation of this cost.

The function $stc(i)$ represents the stability cost of activity i in the original method introduced by [Van de Vonder et al. \(2008\)](#). Let us introduce the function $\hat{stc}(i)$ as the Modified stability cost of activity i . We compute $\hat{stc}(i)$ as follows:

$$\begin{aligned} \hat{stc}(i) &= stc(i) \text{ if } i \neq n+1 \\ \hat{stc}(n+1) &= stc(n+1) + w_b S_{n+1} \end{aligned}$$

All steps in our modified STC method are exactly the same as in the original method, except that we use $\sum_{i \in N} \hat{stc}(i)$ instead of $\sum_{i \in N} stc(i)$ as the total stability cost.

4.4.2 Modified Model 2 (M-Model 2)

Model 2 is not directly suitable to be compared with any conventional proactive and reactive solution, since its optimal policies may include deadends. In the modified version of Model 2, once we reach a deadstate, instead of inducing M , we run a very limited simulation to compute an estimated cost of reactions for the remainder of the execution of the project. Each simulation only consists of 10 randomly chosen realizations. During the simulation, all infeasibilities are resolved by RP-SGS where EBST is the scheme's priority rule.

4.4.3 M-Model 2 vs. CONV

We compare CONV and M-Model 2 where $\mathbf{S}^{init} = SMP$ and $|\mathbf{S}| = 1000$. We apply the following experiment: we run both methods for each combination of parameters $(\lambda, w_b, w_r, \alpha)$ where $\lambda = 1, 1.1$ or 1.2 , $w_b = 25$ or 50 , $w_r = 0, 50, 100$ or 200 and $\alpha = 1.1, 1.2$ or 1.3 . To incorporate α in M-Model 2, we limit the choice of the baseline schedule and replace (19) by

$$S^{\text{base}} = \arg \min_{S \in \mathbf{S}_\alpha} \{c_2(S, 0, \emptyset, 0) + w_b S_{n+1}\} \quad (12)$$

w_r	λ	$w_b = 25$			$w_b = 50$		
		$\alpha = 1.1$	$\alpha = 1.2$	$\alpha = 1.3$	$\alpha = 1.1$	$\alpha = 1.2$	$\alpha = 1.3$
0	1	-4.84	-3.79	-3.79	-2.70	-2.62	-2.62
	1.1	-4.69	-2.97	-2.97	-2.66	-2.47	-2.47
	1.2	-2.93	-0.27	-0.27	-1.24	-0.75	-0.75
50	1	-0.66	1.77	1.77	-0.74	-0.51	-0.51
	1.1	-0.26	3.08	3.08	-0.23	0.27	0.27
	1.2	1.48	5.82	5.83	1.40	2.33	2.33
100	1	3.85	7.98	8.01	2.06	2.71	2.71
	1.1	4.17	9.15	9.20	2.62	3.66	3.66
	1.2	5.77	11.56	11.64	4.19	5.73	5.73
200	1	12.20	18.40	18.70	7.93	9.69	9.70
	1.1	12.25	19.18	19.59	8.34	10.60	10.61
	1.2	13.37	20.88	21.36	9.65	12.42	12.44

Table 7: Average percent deviation of M-Model 2 from CONV for different values of parameters λ, w_b, w_r and α

w_r	$w_b = 25$			$w_b = 50$		
	$\alpha = 1.1$	$\alpha = 1.2$	$\alpha = 1.3$	$\alpha = 1.1$	$\alpha = 1.2$	$\alpha = 1.3$
0	-2.13	-1.80	-1.80	-0.41	-0.36	-0.36
50	1.08	2.90	2.90	0.34	0.51	0.51
100	5.24	9.36	9.42	2.84	3.42	3.42
200	13.60	20.26	20.71	8.46	10.14	10.15

Table 8: Average percent deviation of Model 3 from CONV for different values of parameters w_b, w_r and α and fixed value of parameter $\lambda = 1$

where $\mathbf{S}_\alpha = \{S \in \mathbf{S} | S_{n+1} \leq \delta\}$.

Table 7 reports the results of this experiment. Note that the numbers associated to the settings where $w_b = 50$ and $w_r = 0$, for which our assumption $w_b \leq w_{n+1,0} + w_r$ is violated ($w_{n+1,0}$ is set to be 38), are printed in bold. We notice that M-Model 2 performs worse than CONV when $w_r = 0$. However, we observe that by increasing the cost w_r , M-Model 2 tends to outperform CONV. Similarly, by increasing λ , the average percent deviation of M-Model 2 from CONV is also increased.

4.4.4 Model 3 vs. CONV

We compare CONV and Model 3 where $\mathbf{S}^{init} = SMP$ and $|\mathbf{S}| = 1000$. We apply a similar experiment to the one in Section 4.4.3 except that we fix $\lambda = 1$. To incorporate α in Model 3, we limit the choice of the baseline schedule and replace (11) by

$$S^{\text{base}} = \arg \min_{S \in \mathbf{S}_\alpha} \{c_3(cu^0, co_{(S,0)}, 0, 0) + w_b S_{n+1}\}. \quad (13)$$

Table 8 reports the results of this experiment. Similarly to Table 7, the number associated to the settings where $w_b = 50$ and $w_r = 0$ are printed in bold. In terms of average percent deviation from CONV, Model 3 clearly outperforms M-Model 2. We notice that Model 3 also performs worse than CONV when $w_r = 0$ and performs better than CONV when $w_r > 0$. We must acknowledge that although Model 3 outperforms M-Model 2 in all settings and CONV in most of the settings,

w_r	$w_b = 25$			$w_b = 50$		
	$\alpha = 1.1$	$\alpha = 1.2$	$\alpha = 1.3$	$\alpha = 1.1$	$\alpha = 1.2$	$\alpha = 1.3$
0	-0.33	-0.19	-0.19	0.47	0.52	0.52
50	2.36	3.96	3.96	0.91	1.08	1.08
100	5.85	10.11	10.20	3.11	3.77	3.77
200	13.28	20.47	21.05	8.52	10.73	10.75

Table 9: Average percent deviation of Model 4 from CONV for different values of parameters w_b, w_r and α and fixed value of parameter $\lambda = 1$

Figure 11: The precedence graph for the instance of the counterexample

it also requires much more computational resources than both M-Model 2 and CONV, especially when a positive fixed cost is induced with each reaction.

4.4.5 Model 4 vs. CONV

Similarly to the previous comparisons, we compare CONV and Model 4 where $\mathbf{S}^{init} = SMP$ and $|\mathbf{S}| = 1000$. We stop the algorithm of Model 4 if it reaches 200 million states (which requires around 25 GB of memory to store the states and the arcs). Due to lack of memory in the laptop machine, the computational results in this section were obtained on a server machine with Intel Xeon CPU E5-2699 2.3 GHz (2 processors), 256GB of RAM and running under Windows server 2012 R2. To incorporate α in Model 4, we limit the choice of the baseline schedule and replace (24) by

$$S^{\text{base}} = \arg \min_{S \in \mathbf{S}_\alpha} \{c_4(cu^0, co_{(S,0)}, 0, 0) + w_b S_{n+1}\}. \quad (14)$$

Table 9 reports the results of this experiment. Similarly to Table 7 and Table 8, the numbers associated to the settings where $w_b = 50$ and $w_r = 0$ are printed in bold. In terms of average percent deviation from CONV, Model 4 clearly outperforms both M-Model 2 and Model 3. Model 4 performs slightly worse than CONV when $w_r = 0$ and $w_b = 25$ and performs better elsewhere, especially when $w_r \geq 50$.

5 Discussion

In this research, we only consider PR-policies that dictate certain transitions among schedules when conflicts occur. Such PR-policies are called *conflict-based* PR-policies. It might be wrongfully perceived that reactions are only necessary when conflicts occur. We find that reactions can be beneficial even if no conflict occurs. To support our finding, we provide a counterexample where a *non-conflict-based* PR-policy is a better solution than the optimal solution to P. Notice that, in our research, Π only contains conflict-based PR-policies.

Consider the following instance of P. We are given a project with three non-dummy activities. The set of activities is given as $N = \{0, 1, 2, 3, 4\}$ where activities 0 and 4 are dummy start and dummy end activities, respectively. We only have one resource type with availability one. Each activity needs exactly one resource unit to be executed. Figure 11 shows the precedence relations among the activities of this instance. The set $\mathfrak{P} = \{\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4\}$ for this instance contains only four realizations that are given as follows:

$$\begin{array}{ll}
\mathbf{p}_1 = (0, 1, 1, 2, 0) & \pi(\tilde{\mathbf{p}} = \mathbf{p}_1) = 0.21 \\
\mathbf{p}_2 = (0, 1, 2, 2, 0) & \pi(\tilde{\mathbf{p}} = \mathbf{p}_2) = 0.09 \\
\mathbf{p}_3 = (0, 2, 1, 2, 0) & \pi(\tilde{\mathbf{p}} = \mathbf{p}_3) = 0.49 \\
\mathbf{p}_4 = (0, 2, 2, 2, 0) & \pi(\tilde{\mathbf{p}} = \mathbf{p}_4) = 0.21
\end{array}$$

For this counterexample we assume $f(\Pi, l) = f(\Pi, l)$. We set the parameters $w_{0,k} = w_{1,k} = w_{2,k} = 0$, $w_{3,k} = 20$, $w_{4,k} = 40$ for each $k \geq 0$. Let $w_b = 50$, $w_r = 20$, $w_{de} = 1000$.

There is a set $\mathbf{S}^{dom}$ of schedules with which we can construct every dominant PR-policy. For this very small instance, $\mathbf{S}^{dom}$ is very small and only contains the following four schedules:

$$\begin{array}{ll}
S^1 = (0, 0, 1, 2, 4) & S^2 = (0, 0, 1, 3, 5) \\
S^3 = (0, 0, 2, 3, 5) & S^4 = (0, 0, 2, 4, 6)
\end{array}$$

The policy $\Pi^* \in \Pi$, which is given below, happens to be the optimal PR-policy for this instance of P.

$$\Pi^* : \begin{cases} \Phi_{\Pi^*,1} : S^3 & \gamma_{\Pi^*,1} = 0, & f(\Pi^*, 1) = 250 \\ \Phi_{\Pi^*,2} : S^3 \xrightarrow{t=3} S^4 & \gamma_{\Pi^*,2} = 0, & f(\Pi^*, 2) = 250 + 80 \\ \Phi_{\Pi^*,3} : S^3 & \gamma_{\Pi^*,3} = 0, & f(\Pi^*, 3) = 250 \\ \Phi_{\Pi^*,4} : S^3 \xrightarrow{t=3} S^4 & \gamma_{\Pi^*,4} = 0, & f(\Pi^*, 4) = 250 + 80 \end{cases}$$

$$\sum_{l=1}^4 \pi(\tilde{\mathbf{p}} = \mathbf{p}_l) f(\Pi^*, l) = (0.21 + 0.49) \times 250 + (0.09 + 0.21) \times 330 = 274$$

Now consider the following non-conflict-based PR-policy Π' .

$$\Pi' : \begin{cases} \Phi_{\Pi',1} : S^3 \xrightarrow{t=1} S^2 & \gamma_{\Pi',1} = 0, & f(\Pi', 1) = 250 + 20 \\ \Phi_{\Pi',2} : S^3 \xrightarrow{t=1} S^2 & \gamma_{\Pi',2} = 0, & f(\Pi', 2) = 250 + 20 \\ \Phi_{\Pi',3} : S^3 & \gamma_{\Pi',3} = 0, & f(\Pi', 3) = 250 \\ \Phi_{\Pi',4} : S^3 \xrightarrow{t=3} S^4 & \gamma_{\Pi',4} = 0, & f(\Pi', 4) = 250 + 80 \end{cases}$$

$$\sum_{l=1}^4 \pi(\tilde{\mathbf{p}} = \mathbf{p}_l) f(\Pi', l) = 0.49 \times 250 + (0.09 + 0.21) \times 270 + 0.21 \times 330 = 272.8$$

We conclude that there might be non-conflict-based PR-policies with a smaller objective cost than the best conflict-based PR-policy.

6 Summary and future research

In this paper, we introduce a proactive and reactive resource-constrained project scheduling problem with duration uncertainty. The optimal solution of the problem is a policy with the minimum expected combined cost. This expected combined cost is a combination of an expected cost of the baseline schedule, an expected cost of a series of reactions and an expected cost of having no reaction possibility. We propose four different Markov-decision-process models which can solve the problem

over four different classes of policies. The computational results state that Model 2, Model 3 and Model 4 perform reasonably better than the conventional method introduced in [Section 4](#).

In future research, on the one hand, we would like to find solutions with even lower combined cost. Finding a diverse set of schedules which provides good reaction possibilities among schedules is likely to reduce the combined cost of the optimal solution in our models. Therefore, one future research possibility is to search for a wisely selected set of schedules that provides both diversity and similarity among schedules. Also, incorporating non-conflict-based PR-policies in the models boils down to possibly smaller combined costs, and as such is an interesting future research possibility. On the other hand, we would like to increase the speed of our models. Finding new dominance rules to eliminate dominated reaction possibilities might help reducing the computational resources needed to solve our models.

A Appendix

A.1 The recursion of Model 2

We introduce $d_2(S \rightarrow S', t \rightarrow t + c, O, \nu \rightarrow \nu + 1)$ as the expected cost until the end of execution if we decide to transit from (S, t, O, ν) to $(S', t + c, O, \nu + 1)$ in Model 2 and $c_2(S, t, O, \nu)$ as the corresponding expected cost until the end of execution upon leaving each feasible state (S, t, O, ν) .

$$\begin{aligned} d_2(S \rightarrow S', t \rightarrow t + c, O, \nu \rightarrow \nu + 1) &= w_r + \sum_{i \in U(S, t)} (w_{i(\nu+1)} |S'_i + c - S_i|) \\ &\quad + c_2(S', t + c, O, \nu + 1) \end{aligned} \quad (15)$$

Then, the cost associated with the best decision is computed as follows:

$$(S^*, c^*) = \arg \min_{(S', c) \in \Gamma_2(S, t, O)} \{d_2(S \rightarrow S', t \rightarrow t + c, O, \nu \rightarrow \nu + 1)\} \quad (16)$$

$$d_2^*(S, t, O, \nu) = d_2(S \rightarrow S^*, t \rightarrow t + c^*, O, \nu \rightarrow \nu + 1) \quad (17)$$

We introduce the set $\mathcal{F}_2$ and $\mathcal{I}_2$ as the set of all feasible states and the set of all infeasible states for Model 2, respectively. Also, the set of all endstates and the set of all deadstates are represented by $\mathcal{E}_2$ and $\mathcal{D}_2$, respectively. The function $c_2(S, t, O, \nu)$ is computed as follows:

$$\begin{aligned} c_2(S, t, O, \nu) &= \sum_{(S, t', O', \nu) \in \mathcal{F}_2 \setminus \mathcal{E}_2} \pi(S, t \rightarrow t', O \rightarrow O', \nu) * c_2(S, t', O', \nu) \\ &\quad + \sum_{(S, t', O', \nu) \in \mathcal{I}_2 \setminus \mathcal{D}_2} \pi(S, t \rightarrow t', O \rightarrow O', \nu) * d_2^*(S, t', O', \nu) \\ &\quad + \sum_{(S, t', O', \nu) \in \mathcal{D}_2} \pi(S, t \rightarrow t', O \rightarrow O', \nu) * M \end{aligned} \quad (18)$$

We select the baseline schedule (S^{base}) as follows:

$$S^{\text{base}} = \arg \min_{S \in \mathbf{S}} \{c_2(S, 0, \emptyset, 0) + w_b S_{n+1}\} \quad (19)$$

A.2 The recursion of Model 4

We compute the cost until the end of the execution for each decision arc:

$$\begin{aligned} d_4(cu, co \rightarrow co', 0 \rightarrow de', \nu \rightarrow \nu + 1) &= w_r + \\ &\quad \sum_{i \in I} (w_{i(\nu+1)} |\rho_i - (de' + \rho'_i)|) + c_4(cu, co', de', \nu + 1) \end{aligned} \quad (20)$$

The cost associated to the best decision is computed as follows:

$$(co^*, de^*) = \arg \min_{(co', de') \in \Gamma_4(cu, co)} \{d_4(cu, co \rightarrow co', 0 \rightarrow de', \nu \rightarrow \nu + 1)\} \quad (21)$$

$$d_4^*(cu, co, 0, \nu) = d_4(cu, co \rightarrow co^*, 0 \rightarrow de^*, \nu \rightarrow \nu + 1) \quad (22)$$

We introduce the set $\mathcal{F}_4$, $\mathcal{I}_4$ and $\mathcal{E}_4$ as the set of all feasible states, the set of all infeasible states and the set of all endstates in Model 4, respectively. The function $c_4(cu, co, de, \nu)$ is computed as follows:

$$\begin{aligned} c_4(cu, co, de, \nu) = & \quad (23) \\ & \sum_{(cu', co', de', \nu) \in \mathcal{F}_4 \setminus \mathcal{E}_4} \pi(cu \rightarrow cu', co \rightarrow co', de \rightarrow de', \nu) * c_4(cu', co', de', \nu) \\ & + \sum_{(cu', co', 0, \nu) \in \mathcal{I}_4} \pi(cu \rightarrow cu', co \rightarrow co', de \rightarrow 0, \nu) * d_4^*(cu', co', 0, \nu) \end{aligned}$$

We select the baseline schedule (S^{base}) as follows:

$$S^{\text{base}} = \arg \min_{S \in \mathbf{S}} \{c_4(cu^0, co_{(S,0)}, 0, 0) + w_b S_{n+1}\} \quad (24)$$

References

- Abbasi, B., Shadrokh, S., and Arkat, J. Bi-objective resource-constrained project scheduling with robustness and makespan criteria. *Applied Mathematics and Computation*, 180(1):146–152, 2006.
- Akkan, C., Külünk, M. E., and Koças, C. C. Finding robust timetables for project presentations of student teams. *European Journal of Operational Research*, 249(2):560–576, 2016.
- Al-Fawzan, M. A. and Haouari, M. A bi-objective model for robust resource-constrained project scheduling. *International Journal of Production Economics*, 96(2):175–187, 2005.
- Alvarez-Valdes, R. and Tamarit, J. The project scheduling polyhedron: Dimension, facets and lifting theorems. *European Journal of Operational Research*, 67(2):204–220, 1993.
- Ashtiani, B., Leus, R., and Aryanezhad, M. B. New competitive results for the stochastic resource-constrained project scheduling problem: Exploring the benefits of pre-processing. *Journal of Scheduling*, 14(2):157–171, 2011.
- Ballestín, F. and Leus, R. Resource-constrained project scheduling for timely project completion with stochastic activity durations. *Production and Operations Management*, 18(4):459–474, 2009.
- Ben-Tal, A., Goryashko, A., Guslitzer, E., and Nemirovski, A. Adjustable robust solutions of uncertain linear programs. *Mathematical Programming*, 99(2):351–376, 2003.
- Chaari, T., Chaabane, S., Aissani, N., and Trentesaux, D. Scheduling under uncertainty: Survey and research directions. In *Advanced Logistics and Transport (ICALT), 2014 International Conference on*, pages 229–234, 2014.
- Chtourou, H. and Haouari, M. A two-stage-priority-rule-based algorithm for robust resource-constrained project scheduling. *Computers & Industrial Engineering*, 55(1):183–194, 2008.

- Creemers, S. Minimizing the expected makespan of a project with stochastic activity durations under resource constraints. *Journal of Scheduling*, 18(3):263–273, 2015.
- Creemers, S., Leus, R., and Lambrecht, M. Scheduling markovian PERT networks to maximize the net present value. *Operations Research Letters*, 38(1):51–56, 2010.
- Deblaere, F., Demeulemeester, E., and Herroelen, W. Proactive policies for the stochastic resource-constrained project scheduling problem. *European Journal of Operational Research*, 214(2):308–316, 2011.
- Demeulemeester, E. and Herroelen, W. A branch-and-bound procedure for the multiple resource-constrained project scheduling problem. *Management Science*, 38(12):pp. 1803–1818, 1992.
- Demeulemeester, E. and Herroelen, W. New benchmark results for the resource-constrained project scheduling problem. *Management Science*, 43(11):pp. 1485–1492, 1997.
- Demeulemeester, E. and Herroelen, W. Robust project scheduling. *Foundations and Trends in Technology, Information and Operations Management*, 3:201–376, 2011.
- Flyvbjerg, B., Bruzelius, N., and Rothengatter, W. *Megaprojects and Risk: An Anatomy of Ambition*. Cambridge University Press, 2003.
- Gabrel, V., Murat, C., and Thiele, A. Recent advances in robust optimization: An overview. *European Journal of Operational Research*, 235(3):471 – 483, 2014.
- Herroelen, W. Project scheduling: Theory and practice. *Production and Operations Management*, 14(4):413–432, 2005.
- Herroelen, W. and Leus, R. Project scheduling under uncertainty: Survey and research potentials. *European Journal of Operational Research*, 165(2):289–306, 2005.
- Herroelen, W. and Leus, R. The construction of stable project baseline schedules. *European Journal of Operational Research*, 156(3):550–565, 2004.
- Igelmund, G. and Radermacher, F. J. Preselective strategies for the optimization of stochastic project networks under resource constraints. *Networks*, 13(1):1–28, 1983.
- Kolisch, R. and Sprecher, A. PSPLIB - A project scheduling problem library. *European Journal of Operational Research*, 96(1):205–216, 1997.
- Lamas, P. and Demeulemeester, E. A purely proactive scheduling procedure for the resource-constrained project scheduling problem with stochastic activity durations. *Journal of Scheduling*, 2015. to appear.
- Leus, R. *The generation of stable project plans*. PhD thesis, KU Leuven, 2003.
- Leus, R. and Herroelen, W. The complexity of machine scheduling for stability with a single disrupted job. *Operations Research Letters*, 33(2):151–156, 2005.
- Liebchen, C., Lübbecke, M., Möhring, R., and Stiller, S. *Robust and Online Large-Scale Optimization: Models and Techniques for Transportation Systems*, chapter The Concept of Recoverable Robustness, Linear Programming Recovery, and Railway Applications, pages 1–27. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

- Möhring, R., Radermacher, F., and Weiss, G. Stochastic scheduling problems I - General strategies. *Zeitschrift für Operations Research*, 28(7):193–260, 1984.
- Möhring, R., Radermacher, F., and Weiss, G. Stochastic scheduling problems II - Set strategies. *Zeitschrift für Operations Research*, 29(3):65–104, 1985.
- Shapiro, A. A dynamic programming approach to adjustable robust optimization. *Operations Research Letters*, 39(2):83 – 87, 2011.
- Stork, F. *Stochastic resource-constrained project scheduling*. PhD thesis, TU Berlin, 2001.
- Van de Vonder, S., Demeulemeester, E., Herroelen, W., and Leus, R. The use of buffers in project management: The trade-off between stability and makespan. *International Journal of Production Economics*, 97(2):227–240, 2005.
- Van de Vonder, S., Demeulemeester, E., Herroelen, W., and Leus, R. The trade-off between stability and makespan in resource-constrained project scheduling. *International Journal of Production Research*, 44(2):215–236, 2006.
- Van de Vonder, S., Demeulemeester, E., and Herroelen, W. A classification of predictive-reactive project scheduling procedures. *Journal of Scheduling*, 10(3):195–207, 2007.
- Van de Vonder, S., Demeulemeester, E., and Herroelen, W. Proactive heuristic procedures for robust project scheduling: An experimental analysis. *European Journal of Operational Research*, 189(3):723–733, 2008.

FACULTY OF ECONOMICS AND BUSINESS

Naamsestraat 69 bus 3500

3000 LEUVEN, BELGIË

tel. + 32 16 32 66 12

fax + 32 16 32 67 91

info@econ.kuleuven.be

www.econ.kuleuven.be