Proactive Reactive Scheduling in Resource Constrained Projects with Flexibility and Quality Robustness Requirements

Mario Brčić, Damir Kalpić, and Marija Katić

University of Zagreb Faculty of Electrical Engineering and Computing, Croatia
mario.brcic@fer.hr, damir.kalpic@fer.hr, marija.katic@fer.hr

Abstract. This paper presents a new approach to proactive reactive scheduling of stochastic resource-constrained project scheduling problems with known probability distributions of activity durations. To facilitate the search for cost-flexible proactive schedules that are adjustable and incur lower expected cost of future rescheduling, a new family of cost-based flexibility measures is introduced. Under these measures, cost is incurred on each rescheduling while taking into account the temporal distance of changes in the baseline schedule. We propose a new model that describes the integrated approach using the proposed cost-based flexibility measures where, in each stage, reactive scheduling can adjust the baseline schedule to accommodate flexibility and quality requirements. The model is based on bounded stochastic shortest path with finite state and action spaces. The commonly used schedule stability measure is put in the context of proposed family of flexibility measures and contrasted to them in the terms of project execution system properties.

Keywords: project scheduling, risk, proactive reactive scheduling, stochastic dynamic programming, cost-based flexibility

1 Introduction

Stochastic Resource Constrained Project Scheduling Problem (SRCPSP) is a generalization of the classical family of deterministic scheduling problems with complete information [1]. It introduces uncertainty by using random variables to model some of its data. In this paper, we shall focus on uncertain activity durations with known probability distributions. There are three main approaches to solving SRCPSP: predictive, reactive and proactive procedures [1] and combinations of the main approaches.

Predictive approach ignores stochasticity of the problem and uses point estimations, most usually expectation or median, instead of random variables. This has been shown to underestimate project cost/duration [2]. Reactive procedures make scheduling decisions during the project run-time. They can work with baseline schedule, as schedule repair procedures, and without baseline schedule as completely online procedures. In the latter case they see the project as a multi-stage decision making process and dynamically create schedule in stages, using policies. Proactive project scheduling for SRCPSP is interested in creating a baseline schedule of increased robustness to unexpected outcomes (according to the used robustness measure explained below) such as longer than anticipated activity duration. In such a way, it can remain feasible under various conditions. The two most commonly used notions of robustness in proactive scheduling are: quality and stability robustness. Quality robustness pertains to maximizing the probability of completing the project on time. Stability robustness aims to make schedule stable with respect to possible disruptions, so it does not change much during the execution. Solution robustness or schedule stability is justified by examples where several separate entities cooperate on the project and need to synchronize their actions. Also, in cases with in-house project running, schedule stability increases the setup efficiency. Quality and stability are most commonly two competing criteria and problems containing both are bi-objective. However, they are implicitly converted to single-objective problems by parameterization into monetary costs, scalarizing the two objectives into one. If the monetary cost is the only interest, then such approach is valid and we can continue our work with that assumption.

Although protected against some future disruptions, proactive baseline schedule can become infeasible during the execution due to unanticipated disturbances. In that case rescheduling needs to be done. At this point, reactive schedule repair procedures are used [3]. Such a combination of proactive and reactive procedures to SRCPSP is called a proactive-reactive approach. Current rescheduling procedures mostly focus on restoring the schedule feasibility by starting activities with the least rescheduling cost w.r.t. the first baseline and/or they do not produce proactive schedules that are hedged against future unexpected outcomes in the same way as it is done for the baseline schedule [4]. This paper explores the problem of proactive-reactive scheduling of SRCPSP where changes to baseline schedule can be made in advance at a lesser cost than if being done at the activity start, hence yielding a new proactive baseline using all the information available up to that moment.

The main contributions of the paper are:

- A new family of flexibility measures based on realistic assumptions on cost functions. Total idleness is shown to be potentially optimal behaviour in certain situations. A bound is put on the worst-case performance of optimal policy, which ensures the termination.
- A model capturing the aspects of the general problem is presented.
- Commonly used stability measure is put into the relation with the model and compared to our family of flexibility measures.

The organization of this paper is as follows: in section 2, we lay out the overview of the related work done in this area. Section 3 shortly presents the problem and section 4 presents the family of flexibility measures. In section 5, we present the stochastic dynamic programming model and in section 6 we put commonly used stability measure into the relation to our model. Finally, section 7 gives conclusions and future work.

2 Related Work

Authors in [1] have offered the survey of resource-constrained project scheduling under uncertainty. In this section we describe related work in pure reactive approaches based on stochastic dynamic programming and proactive-reactive approaches.

The most influential work on pure reactive based methods is given by Möhring et al. [5, 6]. They modelled a general stochastic scheduling problem with regular performance measures (measures that are non-decreasing in activity completion times) as a stochastic dynamic program. Their theoretical results are built on the fact that the total idleness is non-optimal behaviour for their problem.Stork, based on [5, 6], dealt with different scheduling policy families in [7]. There are several works on pure reactive scheduling [8–10] that use Markov Decision Process (MDP). Tai in [8] used dynamic programming and authors in [9, 10] used reinforcement learning to find solutions. The work listed above focused on regular performance measures and do not use baseline schedules.

Of proactive-reactive approaches, Leus and Herroelen [11] proposed stability measure expressed as the weighted sum of absolute differences between baseline and realized schedule activity start times. This measure is used in the majority of the project scheduling literature [3], including the works listed below. Van de Vonder et al. in [12] describe Starting Time Criticality + Descent (STC+D) heuristic with surrogate measure and simplifying assumptions in approximations for generation of proactive schedules with time buffering on locked resource flows and predefined policy family. Van de Vonder et al. in [4] used robust schedule generation schemes with priority lists in basic sampling approach and with time windows, where point estimates of duration times were used. Deblaere et al. in [13], based on ideas in STC+D, proposed a family of proactive policies that use activity priority list and release times in parameterization. The final schedule is not necessarily resource-feasible, but it minimizes the combination of expected deviation and due date exceeding costs. In all the works above, the policies used are starting activities in every stage of schedule creation, using variants of parallel/serial scheduling schemes, and perform only just-in-time rescheduling of the activities that are about to be run. Lambrechts in his PhD thesis [14] developed a tabu-search based method that does bi-objective optimization for proactive rescheduling with respect to uncertainty of resource availability. The method is, at the same time, keeping the new schedule close to the schedule obtained in the previous phase using the deviation measure. The author used scalarization to transform bi-objective into a single objective problem. This method achieved moderate results [3].

Although there are various interesting approaches to the problem or resource constrained project scheduling under uncertainty, there is no approach that considers scheduling where rescheduling can be done in advance with smaller cost than if done at activity start times. In this paper we give such a proposal where we approach to the problem as a stochastic dynamic program in a similar way as Möhring et al.[5, 6]. However, in order to model proactiveness we allow for a special family of performance measures (not guaranteed to be regular) on SR-CPSP.

3 The Problem Definition

The problem under consideration in this paper is the single mode non-preemptive stochastic resource-constrained project scheduling problem with quality and flexibility robustness requirements, uncertain activity durations and with known probability distribution of activity durations. Let H be the space of all such problems. Each $h \in H$ is a combinatorial optimization problem defined as a tuple $(V, E, p, R, B, D, \delta, c)$. $V = \{0, ...n+1\}$ is a set of $n+2$ activities where 0 and $n+1$ are dummy activities that represent project beginning and end respectively. Let $V' = V \setminus \{0, n+1\}$. Precedence relation between activities is defined as transitive closure of relation $E \subset V \times V$, where 0 precedes and $n+1$ succeeds all other activities in V. Precedence relation must be asymmetric. Let $\Delta(\mathbb{N}_0^{n+2})$ be the space of all discrete probability distributions defined over $\mathbb{N}_0^{n+2}$ with bounded support. $p \in \Delta(\mathbb{N}_0^{n+2})$ is a joint probability distribution of activity durations represented by a random vector d, where p_0 and p_{n+1}, marginal distributions for dummy activities, have all the mass on duration of 0. Also, $\forall a \in V$ either $p_a(0) = 1$ or $p_a(0) = 0$. $R = \{R_1, ..., R_r\}$ defines a set of r renewable resources and $B \in \mathbb{N}^r$ is a vector of resource availabilities. Activity demands on resources are given in the matrix $D \in \mathbb{N}_0^{(n+2)\times r}$, where $(\forall i \in \{a \in V | p_a(0) = 1\}), (\forall r \in R) D_{i,r} = 0$ and $(\forall r \in R)(\forall i \in V) D_{i,r} \leq B_r$. $\delta \in \mathbb{N}_0$ is the project due date. Before defining the objective function c, we need to define some necessary intermediate objects.

Let S be the countable space of all project states, where each state $x \in S$ stores all relevant information about the project during the execution. This information includes the global project time (the time elapsed from the start of the project execution), statuses of activities, durations of finished activities and the current schedule. Schedule is a vector in $\mathbb{N}_0^{n+2}$, where i-th component is the scheduled start time of activity i. The start time of activity 0 is in each schedule equal to 0. In order to extract the schedule from the state, let us define the *schedule extraction function* $L : S \rightarrow \mathbb{N}_0^{n+2}$. Let C be the countable space of all controls (decisions) that control project's execution. Controls start the execution of activities, change the current baseline schedule or do nothing, at any control point of project. As this is a dynamic optimization problem where the objective function c will be defined as expected total cost, where the next project state and stage cost depend only on current state and control, the solution is a randomized Markov *policy* $\mu^* : S \rightarrow \Delta(C)$ [15] in policy space Ξ_{rand}. Let $\Xi_{\text{det}} \subset \Xi_{\text{rand}}$ be the space of all deterministic Markov policies $\mu : S \rightarrow C$. In the rest of the paper we shall focus on deterministic Markov policies.

Let $N : \mathbb{N}_0^{n+2} \times \Xi_{\text{det}} \rightarrow \overline{\mathbb{N}}_0$, i.e. $N(\gamma, \mu)$ for *realized activity duration vector (scenario)* γ and policy μ, be the number of stages where decisions take place before the end of project execution. When it is clear from the context, we omit the dependency and only write N. For each stage $k = 1..N(\gamma, \mu)$ let $x_k^{\gamma,\mu} \in S$ be the project state reached under duration vector γ and policy μ and let $x_k^{\gamma,\mu|t} \in \mathbb{N}_0$ be the global project time of that state. Let $s_k^{\gamma,\mu,i} \in \mathbb{N}_0$ be the scheduled start time of activity i in the schedule $L(x_k^{\gamma,\mu})$. Let $\Pi \subset \Xi_{\text{det}}$ be the space of admissible policies, i.e. all policies that respect 0-lag precedence, resource, non-anticipativity, non-retroactiveness, non-prematureness constraints and condition of project terminability defined below. We search for the solution in the space Π. Let $Z : \Pi \to \mathbb{N}_0$ be the worst-case schedule duration given the problem $h \in H$ and policy $\mu \in \Pi$, i.e. $Z(\mu) = \max_{\gamma \in \text{supp}(p)} s_N^{\gamma,\mu,n+1}$. 0-lag precedence constraints are defined as:

$$s_N^{\gamma,\mu,j} \geq s_N^{\gamma,\mu,i} + \gamma_i, \forall (i,j) \in E, \forall \mu \in \Pi, \forall \gamma \in \text{supp}(p) \ .$$

The set of concurrent activities at timepoint $t \in \mathbb{N}_0$ under policy μ with the vector of activity durations γ is:

$$\Lambda_{t,\gamma,\mu} = \{i \in V | t - s_N^{\gamma,\mu,i} < \gamma_i\} \ .$$

Using the set of concurrent activities, resource constraints are defined as:

$$\sum_{i \in \Lambda_{t,\gamma,\mu}} D_{ij} \leq B_j, \forall t \geq 0, \forall R_j \in R, \forall \mu \in \Pi, \forall \gamma \in \text{supp}(p) \ .$$

Non-anticipativity ensures that for all scenarios $\gamma \in \mathbb{N}_0^{n+2}$, policies $\mu \in \Pi$, and all timepoints $t \in \mathbb{N}_0$, the behaviour of policy μ at any stage k depends only on the history of γ w.r.t. μ up to t. These constraints are formally described in [5]. Non-retroactiveness constraints for all timepoints t disallow rescheduling of activities started at any timepoints $t' < t$. Also, starting activities at any timepoint $t' < t$ or rescheduling starts of not yet started activities to any timepoint $t' < t$ is forbidden. Non-prematureness constraint $\forall t < 0, \forall i \in V$ prohibits start of activity i at timepoint t. Terminability condition means that $(\forall \mu \in \Pi)(\exists M \in \mathbb{N})(\forall \gamma \in \text{supp}(p))$ project execution finishes at least until M under μ and γ. Let $\Pi_{rand} \subset \Xi_{\text{rand}}$ be the set of admissible randomized Markov policies with similar constraints as described above, generalized to the case of randomized policies.

Definition 1. *Function* $c_{\text{d}} : \mathbb{N}_0 \to \mathbb{R}_+$ *is the* quality robustness penalty. *It is defined as* $c_{\text{d}}(x) = \beta_{\text{d}} \cdot \max(0, x - \delta), \beta_{\text{d}} > 0$*, incurred in stages as stage-cost* $c_{\text{d},s}(x) = \beta_{\text{d}} \cdot \mathbf{1}_{x>\delta}$.

Function $c_{\text{s}} : \mathbb{N}_0^{n+2} \times \mathbb{N}_0^{n+2} \times \mathbb{N} \to \mathbb{R}_+$ is a **rescheduling cost function** defined in the next section as part of a cost-based flexibility measure.

The objective function is $c : S \times \Pi \to \mathbb{R}_+$:

$$c(x_{-1}, \mu) = \mathbb{E}^{\mu}_{\mathbf{d} \sim p} \left[\sum_{k=1}^{N(\mathbf{d},\mu)} \left(c_{\text{s}}(L(x_{k-1}^{d,\mu}), L(x_k^{d,\mu}), x_{k-1}^{d,\mu|t}) + c_{\text{d},s}(x_{k-1}^{d,\mu|t}) \right) \right] \quad (1)$$

where x_{-1} is the initial empty state at the timepoint -1 and x_0^{μ} is the initial running schedule state that contains baseline schedule created by the policy μ offline at stage -1 from the empty schedule in state x_{-1}, at no cost. The effects of applied policy control at stage k are first visible in the state at the next stage.

4 Family of Cost-based Flexibility Measures

Introduction of Cost-based Flexibility (CBF) measure enables modelling situations where rescheduling in advance might be opportunistic due to lower costs. This enables search for flexible proactive schedules that are adjustable and incur minimal total costs for rescheduling and due date exceeding.

Let, $\forall i \in V'$, $c_{\mathrm{s},i} : \mathbb{N}_0 \times \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{R}_+$, $c_{\mathrm{s},i}(x,y,t)$, be the **activity rescheduling cost function** defined on each point of the domain. It is monotonically non-decreasing in $|x-y|$ for each constant $\min(x,y)-t$ and for each constant $|x-y|$ it is monotonically non-increasing in positive $\min(x,y)-t$, i.e. the distance of the schedule change from the current timepoint t. Also, $\forall i \forall t\, (x = y \Rightarrow c_{\mathrm{s},i}(x,y,t) = 0)$.

Rescheduling cost function $c_\mathrm{s} : \mathbb{N}_0^{n+2} \times \mathbb{N}_0^{n+2} \times \mathbb{N} \to \mathbb{R}_+$ measures the difference between two successive schedules $L(x_{k-1}^{\gamma,\mu})$ and $L(x_k^{\gamma,\mu})$:

$$c_\mathrm{s}\left(L\left(x_{k-1}^{\gamma,\mu}\right), L\left(x_k^{\gamma,\mu}\right), x_{k-1}^{\gamma,\mu|t}\right) = \sum_{i \in V'} c_{\mathrm{s},i}(s_{k-1}^{\gamma,\mu,i}, s_k^{\gamma,\mu,i}, x_{k-1}^{\gamma,\mu|t}) \ .$$

Definition 2. ***Cost-based flexibility measure*** *is the function* $c_\mathrm{f} : \Pi \times \mathbb{N}_0^{n+2} \to \mathbb{R}_+$ *of the form:*

$$c_\mathrm{f}(\mu,\gamma) = \sum_{k=1}^{N(\gamma,\mu)} c_\mathrm{s}\left(L\left(x_{k-1}^{\gamma,\mu}\right), L\left(x_k^{\gamma,\mu}\right), x_{k-1}^{\gamma,\mu|t}\right) \ . \qquad (2)$$

Lemma 1. $(\forall h \in H)(\forall \mu' \in \Pi_{rand})(\exists \mu \in \Pi) c(x_{-1},\mu) \le c(x_{-1},\mu')$

Definition 3. $\forall h \in H, \forall t \in \mathbb{N}_0$ *period* $[t, t+1)$ *is total idleness period under policy* μ *and vector of activity durations* γ *if and only if* $\Lambda_{t,\gamma,\mu} = \emptyset$.

Lemma 2. $\exists h \in H$ *such that for* $\mu^* \in \Pi, \exists \gamma \in \mathrm{supp}(p)$ *where there is at least one total idleness period.*

Lemma 2 causes the departure from previous theoretical results laid in [5, 6]. We build results that put bound on the amount of total idleness in optimal policy. That enables the creation of model that can be solved using standard methods.

Theorem 1. *For arbitrary problem* $h \in H$, $\exists \mu_r \in \Pi$ *such that* $c(x_{-1},\mu_\mathrm{r}) = c_\mathrm{d}\left(\sum_{i\in V} \max(\mathrm{supp}(p_i))\right)$. *Also,*
$$(\forall \mu \in \Pi)\left(c(x_{-1},\mu) \le c(x_{-1},\mu_\mathrm{r}) \Rightarrow c_\mathrm{d}\left(Z\left(\mu\right)\right) \le \frac{c_\mathrm{d}\left(\sum_{i\in V}\max(\mathrm{supp}(p_i))\right)}{\min_{d\in\mathrm{supp}(p)} p(d)}\right) \ .$$

Corollary 1. *In the case* $\delta < \sum_{i\in V} \max(\mathrm{supp}(p_i))$, *the bound* ζ *on the worst case project duration* $Z(\mu^*)$ *under optimal policy* μ^* *can be uniquely inferred from the cost bound given in Theorem 1 due to the properties of* c_d. *Otherwise, the solution to the problem is trivial and the bound* ζ *is set to* $\sum_{i\in V}\max(\mathrm{supp}(p_i))$.

Corollary 2. *For non-trivial values of* δ, *we can search for optimal policy only in the set* $\{\mu \in \Pi | (\forall \gamma \in \mathrm{supp}(p)) s_N^{\gamma,\mu,n+1} \le \zeta\}$. *Optimal policy* $\mu^* \in \Pi$ *exists.*

The proofs of Lemma 1, Lemma 2, Theorem 1 and Corollary 2 are given in the appendix. The bound in Theorem 1 is loose, depending on the continuous parameter of probability distribution. Stricter bounds can be found using more information about SRCPSP instance at hand. Relying on the Corollary 2, we can model the problem using finite horizon dynamic programming (DP). Also, using finiteness of the worst case performance of the optimal policy, we can model the problem as a stochastic shortest path with finite state and control spaces, using the undiscounted objective function (1) and infinite horizon DP theory.

5 The Model

In this section, we introduce a new stochastic dynamic programming model for the defined problem based on theoretical results from the previous section. The given problem is modelled as a stochastic shortest path (SSP) problem [16] and it is a Markov Decision Process (MDP) with variable number of stages as is the case in the project scheduling.

Definition 4. *Finite Markov Decision Process is a 7-tuple* (S, C, W, U, P, f, g) *where* S *is the finite discrete state space,* C *is the finite discrete control space,* W *is the finite sample space modelling elementary random information we can receive,* $U : S \to 2^C$ *is the control availability function.* $P(\omega \in 2^W | x \in S, u \in C)$ *is the distribution of random information conditioned on the last state and applied control.* $f : S \times C \times W \to S$ *is the state transition function, and* $g : S \times C \to \mathbb{R}$ *is the immediate cost function. Transition probabilities* $T_{i,u,j}$ *between the states* $i, j \in S$ *under applying control* $u \in U(i)$ *can be obtained by using the transition function* f*:* $T_{i,u,j} = P\left\{\{\omega \in W | f(i, u, \omega) = j\} | i, u\right\}$.

To make the state and the control space finite, we are using the upper bound on duration of project execution, Θ. Θ can be determined heuristically, or a conservative upper bound based on Corollary 1 can be used. In the rest of the paper we assume conservative upper bound. Let $O = \{i \in \mathbb{N}_0 | i \leq \Theta\}$.

Let $F : S \to 2^V$ define the set of resource and precedence feasible activities that have not been started yet. Let $F^s(x_k) \subset 2^{F(x_k)}$ be the set of sets of resource and precedence feasible combinations of activities for each state $x_k \in S$.

State Space. Each state $x_k \in S$ is a tuple (v, K, b, t).

- v - r-tuple $(v_1, ..., v_r)$ containing resource availabilities at state x_k, $\forall j \in R$
- K - state information about project activities; $(\forall a \in V \setminus \{0\})$ $K_a = (\phi_a, s_a, d_a)$. Information for activity 0 are not stored as it is under all policies always started at timepoint 0.
  - $\phi_a \in$ {'inactive', 'started', 'finished'} - the status of activity a
  - s_a - start time of activity a in the current schedule. If the activity has not been started yet, s_a is predicted/scheduled start time.
  - d_a – realized duration of activity a, known only if the activity is completed. Otherwise, it is 0 and has no meaning. This information is useful only for activities whose activity durations are not independent.

- $b \in \{\text{'bounded'}, \text{'non-bounded'}\}$ - the status of project state used to terminate all executions of projects surpassing duration of Θ
- t - the current time of the project global clock, measured in discrete units.

Each element of state x_k is marked using the superscript notation, for example $x_k^{\phi_a}$ is the status of activity a in state x_k. The terminal state $x_k \in S$ is the one in which $x_k^{\phi_{n+1}} = \text{'finished'}$ or $x_k^b = \text{'bounded'}$. The initial state x_{-1} has all resources free, all activities inactive, their scheduled start times set to 0, state marked as non-bounded and the global project time set to -1.

Control Space and Available Controls. $C \subset (O \cup \{\text{'start'}, \text{'empty'}\})^{n+1}$ is the set of controls that present decisions for rescheduling of activity scheduled start times and for starting any set of activities. $(\forall \mu \in \Pi)(\forall x_k \in S)\mu(x_k) \in U(x_k)$, where set of available controls in each state has to satisfy next conditions:

1. $(\forall x_k \in S) U(x_k) = \{[u_1, ..., u_{n+1}] | \{i | u_i = \text{'start'}\} \in F^s(x_k)\}$
2. $(\forall i \in V \setminus \{0\})(\forall x_k \in S) \left(x_k^{\phi_i} \neq \text{'inactive'} \Rightarrow u_i(x_k) = \text{'empty'} \right)$
3. $(\forall i \in V \setminus \{0\})(\forall x_k \in S) \left(u_i(x_k) \notin \{\text{'start'}, \text{'empty'}\} \Rightarrow u_i(x_k) \geq x_k^t \right)$
4. $(\forall x_k \in S) \left(x_k^t < 0 \Rightarrow (\forall i \in V \setminus \{0\}) u_i(x_k) \neq \text{'start'} \right)$.

Random Information. The timing of the next stage is $t_{k+1} := t_k + 1$. Sample space is defined as $W = \text{supp}(p)$. Random information distribution function P is based on activity duration probability distribution p, with the probabilities of random information conditioned on durations of finished activities and on current execution times of running activities. All of the information that the conditioning is done upon is contained in the state and last applied control. Non-anticipativity of policies is ensured by the fact that after (state, action) pair (x, u) we receive information only on running activities in $[x^t, x^t + 1)$ where some subset of activities is finished at timepoint $x^t + 1$ and the rest continues with execution, in a similar way as in [5].

Transition Function. The model uses a notion of *post-decision state*. The transition function f is a composition $f = \sigma \circ \psi$ of the post-decision transition function $\psi : S \times C \to S$ and the stochastic transition function $\sigma : S \times W \to S$. The post-decision transition function ψ updates the state with the applied control, but without receiving new random information. The state we transition to after applying ψ is the post-decision state, and no control can be applied to it. The stochastic transition function σ updates the post-decision state with the random information. The result is a new pre-decision state at the next timepoint at which we can apply the new control. Transition function $f = \sigma \circ \psi$ is defined algorithmically as:

function $\phi(x_k, u_k)$
  if $x_k^{\phi_{n+1}} = \text{'finished'} \vee x_k^b = \text{'bounded'}$ **then return** x_k
  $x_k^{u_k} \leftarrow \text{COPY}(x_k)$

if $x_k^t = \Theta \wedge x_k^{\phi_{n+1}} \neq$ 'finished' $\wedge\, u_{k,n+1} \neq$ 'start' **then**
  $x_k^{u_k,b} \leftarrow$ 'bounded'
  return x_k^u
for all $i \in \{i | u_{k,i} =' start'\}$ **do**
  Take up resources in $x_k^{u_k}$ for activity i
  $x_k^{u_k,s_i} \leftarrow x_k^{u_k,t}$, $x_k^{u_k,\phi_i} \leftarrow$ 'started'
for all $i \in \{i | u_{k,i} =' start' \wedge p_i(0) = 1\}$ **do** $x_k^{u_k,\phi_i} \leftarrow$ 'finished'
for all $i \in \{i | u_{k,i} \in O\}$ **do** $x_k^{u_k,s_i} \leftarrow u_{k,i}$
return $x_k^{u_k}$

function $\sigma(x_k^{u_k}, \omega_k)$
  if $x_k^{u_k,\phi_{n+1}} =$ 'finished' $\vee\, x_k^{u_k,b} =$ 'bounded' **then return** $x_k^{u_k}$
  $x_{k+1} \leftarrow \textsc{copy}(x_k^{u_k})$, $x_{k+1}^t \leftarrow x_k^t + 1$
  for all $i \in \{i | i$ finishes in x_{k+1}^t according to $\omega_k\}$ **do**
    release resources in x_{k+1} for i
    $x_{k+1}^{\phi_i} \leftarrow$ 'finished', $x_{k+1}^{d_i} \leftarrow x_{k+1}^t - x_{k+1}^{s_i}$
  return x_{k+1}

Cost Function.

$$g(x_k, u_k) := \begin{cases} 0, & \text{if } x_k^{\phi_{n+1}} = \text{'finished'} \\ & \vee x_k^b = \text{'bounded'} \\ c_\mathrm{s}(L(x_k), L(\psi(x_k, u_k)), x_k^t) + c_{\mathrm{d},s}(x_k^t), & \text{if } x_k^t \geq 0 \\ M, & \text{if } x_k^t = \Theta \wedge u_{k,n+1} \neq \text{'start'} \\ 0, & \text{otherwise} \end{cases}$$

where M is a sufficiently large penalty for not terminating the project before exceeding the bound Θ. Cost bound from Theorem 1 can be used as a conservative basis for setting M.

The proposed model proceeds through all unit timesteps and at each stage searches for optimal controls in high dimensional discrete space, where the dimension depends on the number of activities. Similarly to [13], resource feasibility is not explicitly enforced in the non-started part of schedule. In our model the same holds for precedence feasibility as well.

Since the proposed model is SSP with finite state and control spaces with all admissible policies terminating (some are artificially made proper by bounding and penalizing), there is a unique optimal cost-to-go function $J^* : S \to \mathbb{R}_+$ that satisfies Bellman's optimality equations [16]:

$$J^*(x) = \min_{u \in U(x)} \mathbb{E}_{\omega \sim W(x,u)} \left[g(x,u) + J^* \left(f \left(x, u, \omega \right) \right) \right], \forall x \in S \ . \tag{3}$$

Standard methods for solving MDP such as value iteration, policy iteration or linear programming converge to the solution for this model [16] and can generally be used for solving. The proposed model is nearly acyclic and the solution can be found using simple adaptation to the shortest path algorithm for Directed

Acyclic Graphs (DAG). Such an adaptation sets optimal costs-to-go value of terminal states to 0, resolving the only cycles in the graph, and uses expectations in calculations of distances in stochastic transitions.

6 Stability vs. Cost-based Flexibility

The research based on the schedule stability measure as defined by [11] a priori assumes that the baseline schedule is static and that changes on scheduled activity start times between the creation of the baseline and the realized activity start times are forbidden or of no benefit and are ignored. We consider that the costs due to (in)stability have their root in inflexibilities in the project executing system and that these costs should be lower with increased temporal distance of changes from the current timepoint. That gives us the incentive to "switch" the baseline schedule in order to reduce anticipated costs.Here we present the conditions under which CBF reduces to the stability measure and when the search for the optimal policy can be done in simpler policy subspace.

Definition 5. *$\forall h \in H, \Pi_{\mathrm{st}} \subset \Pi$ is the space of policies that offline, before the start of project execution, create a baseline schedule while online, $\forall \gamma \in \mathrm{supp}(p)$, in each stage only perform rescheduling of activities at their start and leave the non-started part of the schedule unchanged.*

Lemma 3. *$\forall h \in H$ it holds:*

$$(\forall \mu \in \Pi_{\mathrm{st}})(\forall \gamma \in \mathrm{supp}(p)) \left(c_{\mathrm{f}}\,(\mu, \gamma) = \sum_{i \in V'} c_{\mathrm{s},i}(s_0^{\gamma,\mu,i}, s_N^{\gamma,\mu,i}, s_N^{\gamma,\mu,i}) \right) \quad .$$

Theorem 2. *$\forall h \in H, min_{\mu \in \Pi} c(x_{-1}, \mu) = min_{\mu' \in \Pi_{st}} c(x_{-1}, \mu')$ if $(\forall i \in V') c_{\mathrm{s},i}(x, y, t)$ have the following properties:*

1. $(\forall x, y) \min_t c_{\mathrm{s},i}(x, y, t) = c_{\mathrm{s},i}(x, y, \min(x, y))$,
2. $(\forall x, y, z) \left[c_{\mathrm{s},i}(x, y, \min(x, y)) + c_{\mathrm{s},i}(y, z, \min(y, z)) \geq c_{\mathrm{s},i}(x, z, \min(x, z))\right]$,
3. $(\forall x, y) \left[x \leq y \Rightarrow (c_{\mathrm{s},i}(x, y, x) = c_{\mathrm{s},i}(x, y, y))\right]$.

The proofs of Lemma 3 and Theorem 2 are given in the appendix.

Corollary 3. *Let $h \in H$ be such that the expression for c_{f} under Lemma 3, $c_{\mathrm{f,st}}$, has the form of Leus and Herroelen's stability measure. Let h' be the problem identical to h except that c_{f} in the objective function of h' is replaced by $c_{\mathrm{f,st}}$. If h satisfies conditions of the Theorem 2 then $\min_{\mu \in \Pi} c(x_{-1}, \mu) = \min_{\mu' \in \Pi_{\mathrm{st}}} c'(x_{-1}, \mu')$.*

Corollary 3 shows that a priori decision on search in Π_{st} and using stability measure results in no loss of optimality if there is obviously no advantage in rescheduling in advance and if the project executing system is fairly inflexible.

Let $\tau(x, y, t) = \max\{0, \min(x, y) - t\}$. An example of CBF measure is defined by the following rescheduling cost function:

$$c_{\mathrm{s}} \left(L(x_{k-1}^{d,\mu}), L(x_k^{d,\mu}), x_{k-1}^{d,\mu|t} \right) = \sum_{i \in V'} b_i \alpha_i^{\tau(s_{k-1}^{d,\mu,i}, s_k^{d,\mu,i}, x_{k-1}^{d,\mu|t})} \cdot |s_{k-1}^{d,\mu,i} - s_k^{d,\mu,i}| \quad (4)$$

where $b_i \in \mathbb{R}$ is the activity-specific basic cost of rescheduling while $\alpha_i \in (0, 1]$ is activity-specific discount factor. α_i does not model the economic discounting, but the inflexibility of the project execution system included in execution of that activity. For example, the system is inflexible if the discount factor is very close to 1 as there is small or no benefit to reschedulings with advance notice. Obviously, under minimization of the problem that has rescheduling cost function (4) with $\forall i \in V'(\alpha_i > 1)$ we can restrict the search for the solution to Π_{st}. Using Corollary 3, the solution to the problem $h \in H$ with the rescheduling cost function (4), where $\forall i \in V'(\alpha_i = 1)$, can be found by solving the problem h' with the flexibility cost of the form of stability measure, with restriction to search in Π_{st}.

7 Conclusions and Future Work

In this paper, a new approach to proactive-reactive scheduling in SRCPSP has been introduced. To the best of our knowledge, this is the first work that approaches modelling of proactive scheduling with proactive reschedules. There are three main contributions discussed. Firstly, a new family of cost based flexibility measures is proposed in order to measure flexibility robustness. Flexibility measures are integrated into the sequential decision making procedure in order to obtain reactive approach with proactive reschedules. Furthermore, we presented a bounded stochastic shortest path based model with factored state representations that captures important aspects of the given problem. The model is finite MDP with variable number of stages. Standard solving methods, including simple adaptation to the shortest path algorithm for DAGs, could be applied to it using Bellman's optimality equation. Optimal solutions can be obtained only for small projects. The proactive-reactive optimization is done within a single framework of dynamic programming. Our third contribution refers to the commonly used stability measure. We compared it with the proposed family of flexibility measures in the context of proactive scheduling. Schedule stability is shown to be attained under special conditions on the flexibility measure when the project execution system does not show more flexibility with the advance notice or even forbids the advance notice.

Possible future work refers to finding better upper bounds on the worst case schedule duration for optimal policies with bounds depending on discrete parameters of the project. Characterizing the trade-off between the expected performance and different imputed worst case bounds is important for the development of solving procedures. The creation of approximation methods, possibly exploiting near-acyclicity, in order to scale the application scope onto bigger projects is viable venue. For example, approximate dynamic programming can be used on the model with heuristically determined bound on the worst case schedule duration. Finding rescheduling cost function, i.e. sub-elements of flexibility measure, between the consecutive schedules that would balance realistic modelling and computational costs of solving is also an interesting research topic. Finally, research into policy families that would be appropriate for solving problems with special classes of cost-based flexibility measures, could bring the results closer to the application domain.

References

1. Brcic, M., Kalpic, D., Fertalj, K.: Resource constrained project scheduling under uncertainty: A survey. In: Proceedings of the 23rd Central European Conference on Information and Intelligent Systems, 401–409 (2012)
2. Klingel, A.R.: Bias in pert project completion time calculations for a real network. Management Science **13**(4), 194–201 (1966)
3. Demeulemeester, E., Herroelen, W.: Robust Project Scheduling. Now Publishers Inc., Boston (2011)
4. Van de Vonder, S., Ballestin, F., Demeulemeester, E., Herroelen, W.: Heuristic procedures for reactive project scheduling. Computers & Industrial Engineering **52**(1), 11–28 (2007)
5. Möhring, R.H., Radermacher, F.J., Weiss, G.: Stochastic scheduling problems I — general strategies. Zeitschrift für Operations Research **28**(7), 193–260 (1984)
6. Möhring, R.H., Radermacher, F.J.: Introduction to stochastic scheduling problems. In Neumann, P.D.K., Pallaschke, P.D.D., eds.: Contributions to Operations Research. Number 240 in Lecture Notes in Economics and Mathematical Systems, pp. 72–130. Springer Berlin, Heidelberg (1985)
7. Stork, F.: Stochastic resource-constrained project scheduling. PhD thesis, Technical University at Berlin, Berlin, Germany (2001)
8. Tai, C.H.C.: A stochastic project scheduling problem with resource constraints. PhD thesis, University of Missouri-Columbia (1997)
9. Choi, J., Realff, M.J., Lee, J.H.: A Q-learning-based method applied to stochastic resource constrained project scheduling with new project arrivals. International Journal of Robust and Nonlinear Control **17**(13), 1214–1231 (2007)
10. Csáji, B., Monostori, L.: Adaptive stochastic resource control: a machine learning approach. Journal of Artificial Intelligence Research **32**(1), 453—486 (2008)
11. Leus, R., Herroelen, W.: Stability and resource allocation in project planning. IIE Transactions **36**(7), 667–682 (2004)
12. Van de Vonder, S., Demeulemeester, E., Leus, R., Herroelen, W.: Proactive-reactive project scheduling trade-offs and procedures. In Józefowska, J., Weglarz, J., eds.: Perspectives in Modern Project Scheduling. Volume 92, 25–51 Springer US (2006)
13. Deblaere, F., Demeulemeester, E., Herroelen, W.: Proactive policies for the stochastic resource-constrained project scheduling problem. European Journal of Operational Research **214**(2), 308–316 (2011)
14. Lambrechts, O.: Robust project scheduling subject to resource breakdowns. PhD thesis, Faculty of Business and Economics, Katholieke Universiteit Leuven, Leuven, Belgium (2007)
15. Puterman, M.L.: Markov Decision Processes: Discrete Stochastic Dynamic Programming. 1 edn. Wiley-Interscience (1994)
16. Bertsekas, D.P.: Dynamic programming and optimal control, vol.1. 3 edn. Athena Scientific, Belmont, Mass. (2005)

Appendix for Proactive Reactive Scheduling in Resource Constrained Projects with Flexibility and Quality Robustness Requirements

Mario Brčić, Damir Kalpić, and Marija Katić

University of Zagreb Faculty of Electrical Engineering and Computing, Croatia
mario.brcic@fer.hr, damir.kalpic@fer.hr, marija.katic@fer.hr

1 Proofs

Lemma 1. $(\forall h \in H)(\forall \mu' \in \Pi_{rand})(\exists \mu \in \Pi)c(x_{-1}, \mu) \leq c(x_{-1}, \mu')$

Proof. Take arbitrary $h \in H, \mu' \in \Pi_{rand}$.

Let $Z_{rand} : \Pi_{rand} \to \mathbb{N}_0$ be the worst-case schedule duration given the problem $h \in H$ and policy $\mu \in \Pi_{rand}$.

By the properties of Π_{rand}, there exists $M \in \mathbb{N}$ such that $\forall \gamma \in \text{supp}(p)$, μ' terminates project execution at least until time M for all possible sequences of controls that μ can make for γ.

Then, we can obviously search for policy with better or equal value of objective function in the set:

$$\rho' = \{\mu \in \Pi_{rand} | Z_{rand}(\mu) \leq M\}$$

As we know that all policies in ρ terminate project execution up to M and the cost of rescheduling is non-negative through the cost-based flexibility measure, we can restrict our attention from ρ' to

$$\rho = \{\mu \in \rho' | \mu \text{ never does recheduling to timepoints beyond M in states with global project time} \leq M\}$$

Due to reasons stated above:

$$(\forall \mu_1 \in \rho')(\exists \mu_2 \in \rho)c(x_{-1}, \mu_2) \leq c(x_{-1}, \mu_1)$$

From the initial state x_{-1} for all policies in ρ, only members of subset of the states with global project time up to M are participating in the calculation of objective function c as non-zero members in the sum. Using that fact, we model h' as a finite horizon dynamic programming problem h' with $M + 3$ stages, k=-1,0,...,M+1. Problem h' has finite state space:

$$S' = \{x \in S | \text{global project time in x is less than or equal to M+1}\}$$

and finite control space

$$C' = \{c \in C | \text{c does no rescheduling to timepoints beyond M}\}$$

Let:

$$I = \{x \in S' | x^t = M+1 \text{ and the project is not finished at M}\}$$

$M+1$ stages are used for modelling because the effect of starting the activity $n+1$ at timepoint k (activity $n+1$ finishes immediately) is visible only at the next state (at the timepoint $k+1$).

Also, the objective function $c_{h'}$ of problem h' is adapted from the h in a way that there is a penalty for not finishing project execution until the global project time M.

There is a project non-termination cost equal to:

$$Q = \frac{c(x_{-1}, \mu') + 1}{\min_{d \in \text{supp}(p)} p(d)}$$

and a project non-termination cost function $c_Q : S \to \mathbb{R}_+$:

$$c_Q(x) = \begin{cases} Q, \text{ if } x \in I \\ 0, \text{ otherwise} \end{cases}$$

The objective function $c_{h'}$ is defined as:

$$c_{h'}(x_{-1}, \mu) = \mathbb{E}^{\mu}_{\mathbf{d} \sim p} \left[\sum_{k=1}^{M+1} c_s(L(x^{d,\mu}_{k-1}), L(x^{d,\mu}_{k}), x^{d,\mu|t}_{k-1}) + c_d(s^{\gamma,\mu,n+1}_{M+1}) + c_Q(x^{d,\mu}_{M+1}) \right]$$

where the expectation in the above expression is calculated over all activity duration vectors $d \in \text{supp}(p)$ and control sequences that policy $\mu \in \Pi_{rand}$ can make for each of the activity duration vectors.

None of the states in I is reachable from x_{-1} under any of the policies in ρ. $(\forall \mu \in \rho)(\forall \gamma \in \text{supp}(p))$ the project terminates at least until global project time M and for all sequences of decisions that μ can make for γ so the expected value of the sum of quality robustness stage-costs in c is equal to the expected value of quality robustness penalty stated in $c_{h'}$.

Due to reasons stated:

$$(\forall \mu \in \rho)\left(c_{h'}(x_{-1}, \mu) = c(x_{-1}, \mu)\right)$$

Let $\Pi_{h'}$ be the space of admissible deterministic Markov policies that terminate project in the worst case until the global project time M on problem h'. Let $\Pi_{nt} \supset \Pi_{h'}$ be the space of deterministic Markov policies that satisfy all conditions on admissibility except that the condition of project terminability is not neccessarily satisfied and they do not terminate project in the worst case at least until the project time M on problem h'.

At stage k, policies in Π_{nt} reach states x with global project time k. No state is encountered more than once under any policy in Π_{nt}(as in each stage global project time is incremented by 1) so policies in Π_{nt} can be considered stationary as the control depends only on the state. As the described problem h' is finite horizon dynamic programming problem with finite state space and finite control space, then by Theorem 4.4.2. and Proposition 4.4.3. in [1], $(\exists \mu^*_{h'} \in \Pi_{nt})(\forall \mu'' \in \rho) c(x_{-1}, \mu) \leq c(x_{-1}, \mu'')$ - exists optimal deterministic Markov policy on h'.

We will show that $\mu^*_{h'} \in \Pi_{h'}$, i.e. that $\mu^*_{h'}$ terminates project in the worst case at least until the global project time M.

$\forall \mu \in \Pi_{nt} \setminus \Pi_{h'}, \exists \gamma \in \text{supp}(p)$ such that the project is not finished until the global project time M. Then,

$$\begin{aligned} c_{h'}(x_{-1}, \mu) &= \mathbb{E}^{\mu}_{\mathbf{d} \sim p} \left[\sum_{k=1}^{M+1} c_s(L(x^{d,\mu}_{k-1}), L(x^{d,\mu}_k), x^{d,\mu|t}_{k-1}) + c_d(s^{\gamma,\mu,n+1}_{M+1}) + c_Q(x^{d,\mu}_{M+1}) \right] \\ &\geq p(\gamma) * Q \\ &= \frac{p(\gamma)}{\min_{d \in \text{supp}(p)} p(d)} \cdot (c(x_{-1}, \mu') + 1) \\ &> c(x_{-1}, \mu') \end{aligned}$$

From this it follows that $\mu^*_{h'} \in \Pi_{h'}$.

From policy $\mu^*_{h'}$ (optimal on h'), a policy $\mu_h \in \Pi$, defined on h can be constructed through the following expansion:

$$\mu_h(x \in S) := \begin{cases} \mu^*_{h'}(x), & \text{if } x \in S' \setminus I \\ \text{arbitrary admissible control in x,} & \text{otherwise} \end{cases}$$

Let $R_{\mu^*_{h'}}$ be the set of reachable states from x_{-1} participating in the calculation of objective functions $c_{h'}$ under policy $\mu^*_{h'}$. Obviously, $R_{\mu^*_{h'}} \subset S' \setminus I$.

Let R_{μ_h} be the set of reachable states from x_{-1} participating in the calculation of objective functions c under policy μ_h. Obviously, $R_{\mu_{h'}} \subset S' \setminus I$.

Through construction of μ_h it is evident that:

$$R_{\mu^*_{h'}} = R_{\mu_h}$$

Evidently, it holds:

$$c_{h'}(x_{-1}, \mu^*_{h'}) = c(x_{-1}, \mu_h)$$

Also,

$$c(x_{-1}, \mu_h) \leq c(x_{-1}, \mu')$$

□

Lemma 2. *$\exists h \in H$ such that for $\mu^* \in \Pi, \exists \gamma \in \text{supp}(p)$ where there is at least one total idleness period.*

Proof. We shall show this by example. Let us take the next SRCPSP, with V and E given by the graph G(V,E) on Fig.1, where 0 and 3 are dummy activities:

Let the project due date be: $\delta = 4$. Activity duration scenarios: $\gamma_1 = [0,3,2,0]; \gamma_2 = [0,1,2,0]$ with $p(\gamma_1) = 1 - \psi, p(\gamma_2) = \psi$ where $\psi \in (0,1)$.Resources and their availabilities are given by $R = \{R_1\}; B = [1]$. Let resource demands be: $D = [0,1,1,0]^T$. Finally, activity rescheduling cost functions and quality robustness measure:

$$c_{s,1}(x,y,t) = |x-y|, c_{s,2}(x,y,t) = |x-y|, c_d(x) = 10 \cdot \max\{0, x - \delta\}$$

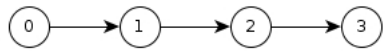

Fig. 1. G(V,E) for the proof of Lemma 2

In the first baseline schedule $(\forall \mu \in \Pi) s_0^{\mu,0} = 0$, by definition. Immediately after the activity 0 is finished, the activity 1 can be started. Let us take arbitrary policy $\mu' \in \Pi$ under which activity $1, \forall \gamma \in supp(p)$, starts in a time $\lambda > 0$ (the same start for all scenarios due to non-anticipativity constraints). Due to precedence constraints, all other activities can start only after the activity 1 has been finished.

Let us construct another policy μ using the policy μ' in the following way:

- in the first baseline schedule: let $s_0^{\mu,i} := max\{0, s_\lambda^{\mu',i} - \lambda\}, \forall i \in V \setminus \{0\}$,
- during the execution: $\forall \gamma \in \{\gamma_1, \gamma_2\}$, let $(\forall i \in V \setminus \{0\})(s_k^{\gamma,\mu,i} = \max\{0, s_{k+\lambda}^{\gamma,\mu',i} - \lambda\})$ be the scheduled start times; if activity i has been started in $s_{k+\lambda}^{\gamma,\mu',i}$ under μ', then start activity i in $s_{k+\lambda}^{\gamma,\mu',i} - \lambda$ under μ

As $\mu' \in \Pi$ and $\forall \gamma \in \{\gamma_1, \gamma_2\}$ μ starts activities with the left shift by λ of activity starts under policy μ' and activity duration scenario γ , μ satisfies precedence and resource constraints. Similar holds for non-retroactiveness constraints. Since $\forall \gamma \in \{\gamma_1, \gamma_2\}$ μ makes decisions only according to decisions of μ' for duration scenario γ and μ' is non-anticipative, μ is also non-anticipative. Hence, $\mu \in \Pi$.

$$\forall \gamma \in \{\gamma_1, \gamma_2\}, s_N^{\gamma,\mu,n+1} = s_N^{\gamma,\mu',n+1} - \lambda$$

$$\forall \gamma \in \{\gamma_1, \gamma_2\}, s_N^{\gamma,\mu,n+1} < s_N^{\gamma,\mu',n+1}$$

Since cost-based flexibility measure c_f in this problem is the sum of monotonically increasing functions in $|x - y|$ (the size of activity start time changes). Considering that $\forall \gamma \in \{\gamma_1, \gamma_2\}$ the changes in start times of activities for the policy μ are smaller than or equal to the corresponding changes in the policy μ':

$$\mathbb{E}_d c_f(\mu, d) \leq \mathbb{E}_d c_f(\mu', d)$$

$c(x_{-1}, \mu) < c(x_{-1}, \mu')$, due to the fact that under the activity duration scenario γ_1, $s_N^{\gamma_1,\mu',n+1} > \delta$ which means that $(\forall \lambda \in \mathbb{N}_1) c_d(s_N^{\gamma_1,\mu,n+1}) < c_d(s_N^{\gamma_1,\mu',n+1})$. Total idleness between activity 0 and the next started activity is suboptimal.

Under the construction given above, the optimal policy for this problem is the one that in the first baseline puts the scheduled start time of activity 1 to time 0, and then starts activity 1 at the time 0. The scheduled start time of activity 3 in the first baseline can be arbitrary since the activity 3 has no rescheduling costs. We have to determine the scheduled start time of activity 2 under optimal policy.

Putting time 0 for scheduled start of activity 2 in the first baseline is suboptimal. Timepoint 0 always results in the increased rescheduling cost in comparison to the timepoint 1, as $\forall \gamma \in \{\gamma_1, \gamma_2\}$ activity 2 cannot start earlier than 1 due to precedence constraints.

Let μ_1 be the policy that creates the first baseline schedule at $[0, 0, 1, 4]$ and starts the activity 1 at the timepoint 0. For activity 1 duration $d_1 = 1$, μ_1 starts the activity 2 at the timepoint 1, and starts activity 3 at the timepoint 3. For activity duration $d_1 = 3$, μ_1 starts the activity 2 at timepoint 3, and starts activity 3 at timepoint 5. Then it holds:

$$c(x_{-1}, \mu_1) = 12 \cdot (1 - \psi)$$

Let μ_2 be the policy that creates the first baseline schedule at $[0, 0, 2, 4]$ and starts the activity 1 at the timepoint 0. For activity 1 duration $d_1 = 1$, μ_2 starts the activity 2 at timepoint 2 and starts the activity 3 at timepoint 4. For activity duration $d_1 = 3$, μ_2 starts the activity 2 at timepoint 3, and starts activity 3 at timepoint 5. Then it holds:

$$c(x_{-1}, \mu_2) = 11 \cdot (1 - \psi)$$

$$\mu_1 \in \operatorname{argmin}\{c(x_{-1}, \mu) | \mu \in \Pi \wedge s_0^{\mu,1} = 0 \wedge s_0^{\mu,2} = 1\}$$

$\forall \mu'' \in \{\mu \in \Pi | c(x_{-1,\mu}) \neq c(x_{-1}, \mu_1) \wedge s_0^{\mu,1} = 0 \wedge s_0^{\mu,2} = 1\}$ it follows that μ'' either has starts of activities 2 and/or 3 delayed (as starts of these activities cannot be done earlier), which incurs the non-negative due date exceeding cost, or the rescheduling from the predicted start of the activity 2 from the timepoint 1 to 3 (in the case of scenario γ_1) is done in a different way which, due to subadditivity of $c_{s,2}$, incurs greater than or equal rescheduling cost as for μ_1. Obviously, selecting the timepoint 1 in the first baseline for the start of activity 2 is suboptimal, meaning that $s_0^{\mu^*,2} > 1$.

Let us take the case of scenario γ_2, where $d_1 = 1$.

Due to non-anticipativity, $s_1^{\gamma_2,\mu^*,2} = s_1^{\gamma_1,\mu^*,2}$. Obviously, $s_0^{\mu^*,2} = s_1^{\gamma_2,\mu^*,2}$ as if the policy μ^* would, at the timepoint 0 - before getting any information, do the rescheduling of the scheduled start of activity 2 to any other timepoint j, it would have greater total cost than the policy μ_j that has all decisions identical to the policy μ^* except that it makes scheduled start of activity 2 in the first baseline equal to j and that would contradict the assumption that μ^* is optimal policy.

As the unit cost of exceeding the due date is bigger than the unit rescheduling cost for activity 2, policies $\mu \in \Pi$ with $s_0^{\mu,2} > 1$, including the μ^*, reschedule the start of the activity 2 to earlier start in order to decrease the due date exceeding cost w.r.t. policy that does not reschedule the scheduled start of activity 2. Let us compare reschedules to the timepoint 1 and to the timepoint 2 at the time 1. Let $s_1^{\gamma_2,\mu^*,2} > 1$ be the scheduled start time of activity 2 when the information of activity 1 finish is available. Policy can do the rescheduling of activity 2 scheduled start to earlier time. Moving the scheduled start of activity 2 earlier, to the timepoint 2 will incur the cost of:

$$-9 \cdot (s_1^{\gamma_2,\mu^*,2} - 2)$$

hence reducing the total cost. Rescheduling to the timepoint 1 would incur cost of

$$-10 \cdot \min\{s_1^{\gamma_2,\mu^*,2} - 2, s_1^{\gamma_2,\mu^*,2} - 1\} + s_1^{\gamma_2,\mu^*,2} - 1 = -9 \cdot (s_1^{\gamma_2,\mu^*,2} - 2) + 1$$

That means rescheduling to timepoint 2 results in lower total cost under γ_2 which means that the rescheduling to timepoint 1 is suboptimal and optimal policy would not be rescheduling scheduled start of activity 2 to it. Hence, that leaves at least one total idleness period, $[1,2)$, under γ_2 and μ^*.

□

Theorem 1. *For arbitrary problem $h \in H$, $\exists \mu_r \in \Pi$ such that $c(x_{-1}, \mu_r) = c_d\left(\sum_{i \in V} \max(\text{supp}(p_i))\right)$.*
$(\forall \mu \in \Pi)\left(c(x_{-1}, \mu) \leq c(x_{-1}, \mu_r) \Rightarrow c_d\left(Z(\mu)\right) \leq \frac{c_d\left(\sum_{i \in V} \max(\text{supp}(p_i))\right)}{\min_{d \in \text{supp}(p)} p(d)}\right)$.

Proof. Take arbitrary $h \in H$.

Let us construct the policy μ_r in the following way:

- Let Γ be a list of activities in V, obtained by the topological sort on $G(V, E)$. Since the $G(V, E)$ is a directed acyclic graph, there exists at least one topological ordering. Let Γ_i be the i-th activity in the ordering Γ.
  μ_r generates the initial baseline schedule $L(x_0^{\mu_r})$ according to the Γ in a recursive way: $s_0^{\Gamma_0} := 0; s_0^{\Gamma_i} := s_0^{\Gamma_{i-1}} + \max(\text{supp}(p_{\Gamma_{i-1}}))$
- During the project execution, policy μ_r starts activities according to their predicted start times in the first baseline $L(x_0^{\mu_r})$ and does not make any other changes.
  - The schedule $L(x_0^{\mu_r})$ is $\forall \gamma \in \text{supp}(p)$ feasible under μ_r. In other words, $\nexists \gamma \in \text{supp}(p)$ such that would yield the predicted start of any activity infeasible.
  - For the completeness sake, policy μ_r in any other state, unreachable from the x_{-1} under μ_r, does start of any nonempty feasible set of activities, if such exists at that state.

Obviously, $\mu_r \in \Pi$. Using the objective function c and the next properties of the rescheduling cost function:

$$c_s(L(x_{k-1}^{d,\mu}), L(x_k^{d,\mu}), x_{k-1}^{d,\mu|t}) = \sum_{i \in V'} c_{s,i}(s_{k-1}^{d,\mu,i}, s_k^{d,\mu,i}, x_{k-1}^{d,\mu|t})$$

$$\forall i \forall t \left(x = y \Rightarrow c_{s,i}\left(x, y, t\right) = 0\right)$$

we can see that the flexibility cost of policy μ_r is 0, and that the total cost is:

$$c(x_{-1}, \mu_r) = c_d(\sum_{i \in V} max(supp(p_i)))$$

Take policy $\mu \in \Pi$ such that it holds:

$$c(x_{-1}, \mu) \leq c(x_{-1}, \mu_r)$$

As by definition of Π, all policies in it terminate, we can for μ write the objective function expression with the quality robustness penalty in the place of sum of all individual quality robustness stage-costs.

Remark: Any policy that has finite objective function value satisfies project terminability condition.

Then,

$$\mathbb{E}^{\mu}_{\boldsymbol{d}\sim p}\left[\sum_{k=1}^{N(\boldsymbol{d},\mu)} c_s(L(x_{k-1}^{d,\mu}), L(x_k^{d,\mu}), x_{k-1}^{d,\mu|t}) + c_d(s_N^{d,\mu,n+1})\right] \leq c_d(\sum_{i\in V} max(supp(p_i)))$$

$$\forall d \in supp(p), p(d)\cdot\left[\sum_{k=1}^{N(\boldsymbol{d},\mu)} c_s(L(x_{k-1}^{d,\mu}), L(x_k^{d,\mu}), x_{k-1}^{d,\mu|t}) + c_d(s_N^{d,\mu,n+1})\right] \leq c_d(\sum_{i\in V} max(supp(p_i)))$$

Let $d' \in \text{argmax}_{d\in\text{supp}(p)} c_d(s_N^{d,\mu,n+1})$. Obviously $Z(\mu) =$, then

$$p(d')\cdot\left[\sum_{k=1}^{N(\boldsymbol{d'},\mu)} c_s(L(x_{k-1}^{d',\mu}), L(x_k^{d',\mu}), x_{k-1}^{d',\mu|t}) + c_d(s_N^{d',\mu,n+1})\right] \leq c_d(\sum_{i\in V} max(supp(p_i)))$$

$$p(d')\cdot c_d(s_N^{d',\mu,n+1}) \leq c_d(\sum_{i\in V} max(supp(p_i)))$$

$$min_{d\in supp(p)} p(d)\cdot c_d(s_N^{d',\mu,n+1}) \leq c_d(\sum_{i\in V} max(supp(p_i)))$$

□

Corollary 2. *For non-trivial values of δ, we can search for optimal policy only in the set $\{\mu \in \Pi | (\forall\gamma \in \text{supp}(p)) s_N^{\gamma,\mu,n+1} \leq \zeta\}$. Optimal policy $\mu^* \in \Pi$ exists.*

Proof. Take arbitrary $h \in H$.

As using Theorem 1 and Corollary 1 we know the bound ζ on worst case project duration for all policies with lesser or equal objective function value than μ_r, obviously we can w.l.o.g. reduce search to $\rho = \{\mu \in \Pi | (\forall\gamma \in \text{supp}(p)) s_N^{\gamma,\mu,n+1} \leq \zeta\}$.

Using the bounded worst-case project duration for all policies in ρ, using the similar procedure as in the proof of Lemma 1 we can construct the finite horizon model h' and find the optimal policy $\mu^*_{h'} \in \Pi_{h'}$.

Using the same expansion as in the proof of Lemma 1 we can from $\mu^*_{h'}$ construct $\mu^* \in \Pi$ that is optimal on problem h. □

Lemma 3. *$\forall h \in H$ it holds:*

$$(\forall\mu \in \Pi_{st})(\forall\gamma \in supp(p))\left(c_f(\mu,\gamma) = \sum_{i\in V'} c_{s,i}(s_0^{\gamma,\mu,i}, s_N^{\gamma,\mu,i}, s_N^{\gamma,\mu,i})\right)$$

Proof. The statement follows from the properties of policies in the space Π_{st} and the property of activity rescheduling cost functions:$\forall i \forall t\, (x = y \Rightarrow c_{s,i}\,(x, y, t) = 0)$.

Take arbitrary $h \in H, \mu \in \Pi_{st}, \gamma \in supp(p)$. By definition,$\forall i \in V$, μ makes at most one change of the activity i scheduled start time during the project execution after the first baseline is created and that change can occur only at the start of activity i.

Take arbitrary $i \in V'$ and let $z^{\gamma,\mu,i}$ be the start time of activity i under policy μ and scenario γ. Let κ be such that $x_{\kappa-1}^{\gamma,\mu|t} = z^{\gamma,\mu,i}$.

$$\begin{aligned}
&\textstyle\sum_{k=1}^{N(\gamma,\mu)} c_{s,i}(s_{k-1}^{\gamma,\mu,i}, s_k^{\gamma,\mu,i}, x_{k-1}^{\gamma,\mu|t}) \\
&= \textstyle\sum_{k\in\{1..N(\gamma,\mu)\}\setminus\{\kappa\}} c_{s,i}(s_{k-1}^{\gamma,\mu,i}, s_k^{\gamma,\mu,i}, x_{k-1}^{\gamma,\mu|t}) + c_{s,i}(s_{\kappa-1}^{\gamma,\mu,i}, s_\kappa^{\gamma,\mu,i}, x_{\kappa-1}^{\gamma,\mu|t}) \\
&= c_{s,i}(s_{\kappa-1}^{\gamma,\mu,i}, s_\kappa^{\gamma,\mu,i}, x_{\kappa-1}^{\gamma,\mu|t}) \\
&= c_{s,i}(s_{\kappa-1}^{\gamma,\mu,i}, s_\kappa^{\gamma,\mu,i}, s_\kappa^{\gamma,\mu,i})
\end{aligned} \tag{1}$$

The third line in (1) was obtained using the fact that the only change of start time could occur at the stage $\kappa-1$, and using the property of activity rescheduling cost functions $\forall i \forall t\, (x = y \Rightarrow c_{s,i}\,(x, y, t) = 0)$ on all other summands. The last line in (1) uses the fact that the activity i was started at that time so $s_\kappa^{\gamma,\mu,i} = x_{\kappa-1}^{\gamma,\mu|t} = z^{\gamma,\mu,i}$.

There was no change in scheduled start time of activity i before the $\kappa - 1$, so $s_{\kappa-1}^{\gamma,\mu,i} = s_0^{\gamma,\mu,i}$. Similarly, there was no change after the $\kappa - 1$ so $s_\kappa^{\gamma,\mu,i} = s_N^{\gamma,\mu,i}$. From this the claim follows. □

Theorem 2. *$\forall h \in H$, $min_{\mu\in\Pi} c(x_{-1}, \mu) = min_{\mu'\in\Pi_{st}} c(x_{-1}, \mu')$ if $(\forall i \in V')c_{s,i}(x, y, t)$ have following properties:*

1. $(\forall x, y) \min_t c_{s,i}(x, y, t) = c_{s,i}(x, y, \min(x, y))$
2. $(\forall x, y, z)\, [c_{s,i}(x, y, \min(x, y)) + c_{s,i}(y, z, \min(y, z)) \geq c_{s,i}(x, z, \min(x, z))]$
3. $(\forall x, y)\, [x \leq y \Rightarrow (c_{s,i}(x, y, x) = c_{s,i}(x, y, y))]$

Proof. Interpretations of properties of functions $c_{s,i}$:

1. cost of rescheduling is the minimal if done just-in-time (no benefit in advance notice)
2. cost of sequence of smaller changes done just-in-time is worse or equal to the cost of total change done made in one step
3. if the rescheduling is done to increase the start time, then the cost of rescheduling at either of the timepoints is equal

$\forall h \in H, \forall x_k \in S$, policy $\mu : S \to C$ can be written as a vector of controls reacting on activities $i \in V \setminus \{0\}$:

$$\mu(x_k) = [(\mu(x_k))_1, ..., (\mu(x_k))_{n+1}] \tag{2}$$

Assume there is $h \in H$ that satisfies conditions of the theorem. Assume that $\mu^* \in \Pi$ is optimal for h, i.e. attains the minimum in objective function. In that case, either $\mu^* \in \Pi_{st}$, and we have trivially proven that the optimal policy is in Π_{st}, or $\mu^* \notin \Pi_{st}$.

Let us assume $\mu^* \notin \Pi_{st}$. Then, there must exist non-empty set $\Delta^{\mu^*} \subseteq supp(p)$ of activity duration scenarios where $\forall d \in \Delta^{\mu^*} \exists a_d \in V$ such that policy μ^* does change in start time between the project time 0 and the actual start of activity a_d. Let $\Delta_j^{\mu^*} \subseteq \Delta^{\mu^*}, j \in V$ be the set of activity duration scenarios where for each $d \in \Delta_j^{\mu^*}$ policy μ^* does change in start time between the project time 0 and the actual start of activity j. Let $A = \{j \in V | \Delta_j^{\mu^*} \neq \emptyset\}$.

Let $a = \min A$. $\forall d \in \Delta_a^{\mu^*}$, let the scheduled start times of the activity a be $(z_1^d, ..., z_{n_d}^d)$, where $n_d > 2$. z_1^d is the scheduled start time of the activity a in the first baseline schedule, each $z_i^d, (1 < i < n_d)$ is subsequent scheduled start time of the activity a after each change before the start of that activity. $z_{n_d}^d$ is the realized start time of activity a under activity duration scenario d and policy μ^*. Let the total rescheduling cost for activity a on the trajectory of states $(x_{-1}, x_0^{\mu^*}, x_1^{d,\mu^*}, ..., x_{N(d,\mu^*)}^{d,\mu^*})$ made under activity duration scenario d and applied decisions of policy μ^* be the sum of all costs for reschedulings:

$$\begin{aligned}\textstyle\sum_{k=2}^{n_d} c_{s,a}(z_{k-1}^d, z_k^d, t_{z_k^d})) &\geq \textstyle\sum_{k=2}^{n_d} c_{s,a}(z_{k-1}^d, z_k^d, min(z_{k-1}^d, z_k^d)) \\ &\geq c_{s,a}\left(z_1^d, z_{n_d}^d, min(z_1^d, z_{n_d}^d)\right) \\ &= c_{s,a}(z_1^d, z_{n_d}^d, z_{n_d}^d)\end{aligned} \tag{3}$$

where $t_{z_k^d}$ is the project time at which the change from z_{k-1}^d to z_k^d occurred.

The first inequality in (3) is due to the property 1. The second inequality is due to successive application of the property 2. Regarding the equality, either $z_{n_d}^d = min(z_1^d, z_{n_d}^d)$, and the equality is trivial, or $z_1^d < z_{n_d}^d$ when it holds by the property 3.

Let μ' be the policy such that $\forall d \in \Delta_a^{\mu^*}$ makes for the activity a only decisions $z_1^d, z_{n_d}^d$ at the respective times $0, z_{n_d}^d$, where the activity a is started at $z_{n_d}^d$, and $\forall j \in V \setminus \{0, a\}$ decisions are identical to those that μ^* makes. $\forall b \notin \Delta_a^{\mu^*}$, μ' makes the identical decisions as μ^*. Since μ^* is admissible, it follows that μ' must be admissible as well. $\forall d \in \Delta_a^{\mu^*}$, the total rescheduling cost under μ^* for activity a is the leftmost sum in (3) while the total rescheduling cost under μ' is the expression after the equality in (3).

In that case, $\forall d \in \Delta_a^{\mu^*}$ the total cost of rescheduling for all activities on the state trajectory under μ' is:

$$\textstyle\sum_{k=1}^{N(d,\mu')}(\sum_{i \in V' \setminus \{a\}} c_{s,i}(s_{k-1}^{d,\mu',i}, s_k^{d,\mu',i}, x_{k-1}^{d,\mu'|t}) + c_{d,s}(x_{k-1}^{d,\mu'|t})) + c_{s,a}(s_0^{\mu',a}, s_N^{d,\mu',a}, s_N^{d,\mu',a}) \leq$$
$$\textstyle\sum_{k=1}^{N(d,\mu^*)}(\sum_{i \in V' \setminus \{a\}} c_{s,i}(s_{k-1}^{d,\mu^*,i}, s_k^{d,\mu^*,i}, x_{k-1}^{d,\mu^*|t}) + c_{d,s}(x_{k-1}^{d,\mu^*|t})) + \sum_{k=1}^{N(d,\mu^*)} c_{s,a}(s_{k-1}^{d,\mu^*,a}, s_k^{d,\mu^*,a}, x_{k-1}^{d,\mu^*|t})$$

From this, it follows that

$$c(x_{-1}, \mu') \leq c(x_{-1}, \mu^*) \tag{4}$$

Let $A := A \setminus \{a\}$. If $A = \emptyset$, then $\Delta^{\mu'} = \emptyset$ and $\mu' \in \Pi_{st}$ which proves the optimal policy is in Π_{st}.

If $A \neq \emptyset$, similar argument is repeated on μ' as for μ^*, by replacing μ^* with μ' and taking $a = \min A$. In each iteration of such a procedure, $\exists j \in A, \forall d \in \Delta_j^{\mu^*}$, controls in subsequently constructed policy are changed for activity j in a way that start times for that activity are no longer rescheduled between the first baseline and the start of that activity. Furthermore, $|V|$ is finite and after at most $|V|$ iterations we will obtain policy $\mu_{st}^* \in \Pi_{st}$ such that $\Delta^{\mu_{st}^*} = \emptyset$. Also, at each step of such a procedure by (4) we obtain at least as good policy as in the previous iteration, so the final policy μ_{st}^* has the minimal cost among the admissible policies and it is in the Π_{st}. As under these conditions we can always find the optimal policy within Π_{st}, the statement of the theorem holds. Search for optimal policy can w.l.o.g. be done only in the Π_{st}. $\square$

References

1. Puterman, M.L.: Markov Decision Processes: Discrete Stochastic Dynamic Programming. 1 edn. Wiley-Interscience (1994)