



HAL
open science

The Anchor-Robust Project Scheduling Problem

Pascale Bendotti, Philippe Chrétienne, Pierre Fouilhoux, Adèle Pass-Lanneau

► **To cite this version:**

Pascale Bendotti, Philippe Chrétienne, Pierre Fouilhoux, Adèle Pass-Lanneau. The Anchor-Robust Project Scheduling Problem. *Operations Research*, 2019, 10.1287/opre.2022.2315 . hal-02144834

HAL Id: hal-02144834

<https://hal.science/hal-02144834>

Submitted on 31 May 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

The Anchor-Robust Project Scheduling Problem

Pascale Bendotti^{1,2} Philippe Chrétienne² Pierre Fouilhoux² Adèle Pass-Lanneau^{1,2}

¹EDF R&D, 7 boulevard Gaspard Monge, 91120 Palaiseau, France

²LIP6, CNRS, Sorbonne Université, 5 place Jussieu, 75005 Paris, France

{pascale.bendotti, adele.pass-lanneau}@edf.fr, {philippe.chretienne, pierre.fouilhoux}@lip6.fr

May 31, 2019

Abstract

A project scheduling framework is considered where processing times of jobs lie in some uncertainty set. The decision maker needs to compute a baseline schedule in advance, while guaranteeing that some jobs may not be rescheduled later. A subset of jobs is said to be anchored with respect to a baseline schedule if for any realization of processing times in the uncertainty set, the baseline schedule can be repaired in a second stage without changing the starting times of anchored jobs. Each job has an anchoring weight. The Anchor-Robust Project Scheduling Problem (AnchRobPSP) is to find in a first stage a baseline schedule satisfying a given deadline, and a subset of anchored jobs with maximum total weight. AnchRobPSP is considered for several uncertainty sets, such as box or budgeted uncertainty set, and dedicated graph models are presented. AnchRobPSP is shown to be NP-hard even for budgeted uncertainty. Polynomial or pseudo-polynomial algorithms are devised for box uncertainty and special cases of budgeted uncertainty. Numerical experiments for AnchRobPSP under budgeted uncertainty are presented. AnchRobPSP solutions are compared to those of state-of-the-art robust techniques. Finally it is shown how to achieve a trade-off between the number of anchored jobs and the makespan of the baseline schedule.

Keywords: Project scheduling, robust optimization, rescheduling, anchored decisions.

1 Introduction

In project management, it is crucial to ensure the availability of equipment or staff, thus requiring the computation of a schedule ahead of time. When the schedule is about to start, some jobs may take longer to process than expected, hence the schedule may have to be modified accordingly. However, in the context of an industrial complex project, rescheduling jobs might be difficult or costly, hence the decision maker wants to guarantee the starting

times of the jobs that are the most difficult or expensive to reschedule. In this work we investigate how to integrate this criterion in the choice of an initial schedule.

The scope of this work is project scheduling where a set of jobs $J = \{1, \dots, n\}$ must be scheduled while respecting precedence constraints, represented by a directed acyclic graph G . The vertex-set of G is $\bar{J} = J \cup \{s, t\}$ where s (resp. t) is a predecessor (resp. successor) of all jobs, representing the beginning (resp. the end) of the schedule. Each job $i \in J$ has a processing time $p_i \in \mathbb{R}_+$, and $p_s = 0$ by convention. Given a vector $p \in \mathbb{R}_+^J$, let $G(p)$ be the weighted digraph obtained from G by weighting every arc (i, j) with p_i . A schedule of $G(p)$ is a vector of starting times $x \in \mathbb{R}_+^{\bar{J}}$ such that $x_j - x_i \geq p_i$ for every arc (i, j) of $G(p)$, $x_s = 0$ and $x_t = \max_{i \in J} x_i + p_i$. Note that x_t is the *makespan* of the schedule.

We consider a 2-stage framework where the processing times of jobs are uncertain. At first stage, the decision maker has an instance $G(p)$ to solve, called *baseline instance*. The project is given a deadline $M \geq 0$, and a *baseline schedule* is defined as a schedule of $G(p)$ with makespan at most M . At second stage, the instance is $G(p + \delta)$ for some $\delta \in \mathbb{R}_+^J$, where δ_i is the *disruption* on the processing time of job i . As done classically in robust optimization, we consider that the decision maker wants to hedge against a family $\{G(p + \delta)\}_{\delta \in \Delta}$ of second-stage instances, where the set Δ is the *uncertainty set*. Given a baseline schedule x^0 , a subset H of jobs is *anchored* with respect to x^0 if all jobs in H are guaranteed to be executed at their starting time in x^0 whatever the realization $\delta \in \Delta$, i.e., for every $\delta \in \Delta$, there exists a schedule x^δ of $G(p + \delta)$ such that $x_i^0 = x_i^\delta$ for every $i \in H$. Each job $i \in J$ has an *anchoring weight* $a_i \in \mathbb{R}_+$, and the objective of the decision maker is to maximize the total weight $a(H) = \sum_{i \in H} a_i$. Given a baseline instance $G(p)$, an uncertainty set Δ , anchoring weights $a \in \mathbb{R}_+^J$, and a deadline M , the Anchor-Robust Project Scheduling Problem (AnchRobPSP) is to find a baseline schedule x^0 and a subset $H \subseteq J$ anchored with respect to x^0 , so that $a(H)$ is maximized.

In practice in large projects a baseline schedule is indeed computed in advance. An example is the planning of maintenance operations for power plants operated by EDF, the French electricity supplier. All operations must be completed during the planned shut-down period of the plant. Anchored jobs are desirable from an operational point of view. For some jobs, the decision maker wants to make the right decision at first try, i.e., schedule the job at a starting time so that it will hopefully not be rescheduled later. This is the case for example if some of the maintenance jobs are achieved by a subcontractor, who would charge extra fees when the jobs are rescheduled. It may also be necessary to anchor jobs that involve a workforce with very specific skills, and thus with limited availability. In this case, anticipation is necessary to ensure that this workforce will actually be available in due time. The anchoring weight a_i represents the extra cost when job i is rescheduled in second stage, while anchored jobs in a solution of AnchRobPSP are jobs that incur no extra cost.

In this work several uncertainty sets are considered. *General uncertainty sets* are any $\Delta \subseteq \mathbb{R}_+^J$. *Upper-bounded uncertainty sets* are sets for which an upper bound δ_i^+ on the increase of the processing time for each job i is known. *Box uncertainty sets* are cartesian products of intervals. *Budgeted uncertainty sets* are defined using a budget Γ on the number of disruptions that may occur simultaneously. Note that the latter two are upper-bounded uncertainty sets of practical interest.

This work fits within the scope of robust optimization, i.e., optimization where some parameters lie in an uncertainty set. Different robust approaches have been proposed since the work of Soyster (1973). These approaches are based on the choice of an uncertainty set and on the design of a deterministic formulation, referred to as *robust counterpart* (see, e.g., (Bertsimas and Sim (2004))). On the one hand, the uncertainty set must be “realistic”, in the sense that it must contain relevant realizations. On the other hand, it must avoid the pitfall of *over-conservatism*, i.e., find not too expensive solutions. Bertsimas and Sim (2004) introduced budgeted uncertainty sets and obtained tractable robust counterparts of combinatorial problems. They showed that the uncertainty budget can be used to control the *price of robustness*, i.e., the increase of the cost of a robust solution w.r.t. the minimum cost of a solution with no uncertainty. As pointed out in the survey from Gabrel et al. (2014), the robust literature proposes *robust-static problems*, where all decisions are made before uncertainty realization, and *robust 2-stage problems*, where some decisions can be made in a second stage after the uncertainty is revealed. AnchRobPSP is thus a robust 2-stage problem, with a specific objective function involving the subset of anchored jobs. Note that AnchRobPSP can be related to the *recoverable robustness* of Liebchen et al. (2009). They introduced a general definition of robust 2-stage problems where the second-stage problem is solved by a given recovery algorithm, e.g., changing a fixed number of decisions.

In project scheduling, several approaches have been proposed to account for uncertainty, see (Herroelen and Leus (2002)) for a survey. When processing times of jobs are uncertain, the robust-static approach is to schedule all jobs to their worst-case processing times. It leads to a schedule with a very large makespan, which is likely to be unapplicable in practice. A robust 2-stage scheduling problem proposed in (Minoux (2007b)) looks for the smallest makespan that can be guaranteed under budgeted uncertainty. This problem is efficiently solvable (Minoux (2007a), Minoux (2007b)). It is worth noticing that this robust 2-stage problem involves no first-stage schedule. The operational need for a precomputed baseline schedule and a feature for maintaining decisions from one stage to the other was emphasized in (Herroelen and Leus (2004)), under the name of *stability of schedules*. The authors studied a stochastic framework where the objective is to minimize the expected value of the total difference of starting times of jobs. The *anchoring level* was introduced in (Bendotti et al. (2017)) as a combinatorial measure of the difference between two schedules. It is defined as

the number of identical starting times between the schedules. A so-called proactive problem was defined to compute a baseline schedule at first stage. This proactive problem is close to recoverable robust timetabling problems studied namely for railway applications (Liebchen et al. (2009), D’Angelo et al. (2011)). Note that a major difference between AnchRobPSP and recoverable robust problems is that in AnchRobPSP the anchored set H is determined at first stage, while in the latter the set of jobs with unchanged starting times depends on the disruption. Indeed, in practice it may be useful for the decision maker to know not only how many jobs will keep the same starting times, but also which ones.

The aim of this work is to provide an in-depth study of AnchRobPSP and to illustrate its relevance for anchoring decisions in project scheduling.

1. A first contribution is a formulation of AnchRobPSP using graph models. We provide two graph models for the problem: the first one is applicable to general uncertainty sets, while the second one is dedicated to budgeted uncertainty sets. We derive a compact MIP formulation for budgeted uncertainty. These graph models are shown to be convenient tools to exhibit hard cases and to devise algorithms.
2. We give complexity results and algorithms for AnchRobPSP under the considered uncertainty sets. For box uncertainty sets, we show that AnchRobPSP is solvable in polynomial time. For upper-bounded uncertainty sets, we show that AnchRobPSP is co-NP-hard. For budgeted uncertainty sets, we show that the decision version of AnchRobPSP is NP-complete, even for fixed budget and when the precedence graph is just a path. We exhibit several polynomial cases. For series-parallel precedence graphs we propose an algorithm which is pseudo-polynomial for fixed budget. For an uncertainty set and its convex hull, we show that the feasible solutions of an instance of AnchRobPSP are the same. We establish that AnchRobPSP is also NP-hard when the uncertainty set is a polytope.
3. Finally we investigate the numerical resolution of AnchRobPSP for budgeted (and box) uncertainty sets. We present insights into how solutions are impacted by the uncertainty budget. We highlight that AnchRobPSP can be used as a tool for achieving a trade-off between anchoring criterion and makespan, which is complementary to existing robust models from the literature.

In Section 2, the graph models for AnchRobPSP are presented. In Section 3, hard cases of AnchRobPSP are exhibited for upper-bounded and budgeted uncertainty. In Section 4, polynomial or pseudo-polynomial algorithms for box uncertainty or special cases of budgeted uncertainty are given. In Section 5, numerical experiments are presented.

2 Graph Models for the Anchor-Robust Project Scheduling Problem

Let \tilde{G} be a directed acyclic graph with vertex-set $\bar{J} = J \cup \{s, t\}$ and let $\tilde{G}(\pi)$ be the weighted digraph obtained by adding arc-weight $\pi_{ij} \in \mathbb{R}_+$ to every arc (i, j) of \tilde{G} . Given a path P in $\tilde{G}(\pi)$, the length of the path P in $\tilde{G}(\pi)$ is denoted by $\ell_{\tilde{G}(\pi)}(P)$. Given two vertices $i, j \in \bar{J}$, let $L_{\tilde{G}(\pi)}(i, j)$ be the length of the longest i - j path in $\tilde{G}(\pi)$ (or $-\infty$ if there is no i - j path in $\tilde{G}(\pi)$). A *schedule* of $\tilde{G}(\pi)$ is a vector $x \in \mathbb{R}_+^{\bar{J}}$ such that $x_j - x_i \geq \pi_{ij}$ for every arc (i, j) of $\tilde{G}(\pi)$. Note that a project scheduling instance $G(p)$ is a special case where $\pi_{ij} = p_i$ for every arc (i, j) . It is well-known that there exists a schedule of $\tilde{G}(\pi)$ such that $x_t \leq M$ if and only if $L_{\tilde{G}(\pi)}(s, t) \leq M$. Given $x \in \mathbb{R}_+^{\bar{J}}$ and $H \subseteq J$, let x_H denote the restriction of vector x to H , i.e., $x_H = (x_i)_{i \in H}$. The following lemma is a straightforward generalization of a result from (Bendotti et al. (2017)), and it will be used in the sequel.

Lemma 1. *Given $\tilde{G}(\pi)$, a set $H \subseteq J$ and x_H^0 a vector of \mathbb{R}_+^J , there exists a schedule x of $\tilde{G}(\pi)$ such that $x_H = x_H^0$ if and only if $x_j^0 - x_i^0 \geq L_{\tilde{G}(\pi)}(i, j)$ for every $i, j \in H \cup \{s\}$.*

Let $(G(p), \Delta, a, M)$ be an instance of AnchRobPSP. It is assumed throughout the paper that $L_{G(p)}(s, t) \leq M$, namely, a baseline schedule always exists. Let us now give definitions of the considered uncertainty sets. Let $\Delta \subseteq \mathbb{R}_+^J$. The set Δ is referred to as a *general uncertainty set* when no further condition is imposed. The set Δ is an *upper-bounded uncertainty set* if there exists $\delta^+ \in \mathbb{R}_+^J$ such that for every $\delta \in \Delta$, $\delta_i^+ \geq \delta_i$ for every $i \in J$, i.e., the duration of job i in a second-stage instance is at most $p_i + \delta_i^+$. Note that δ^+ is not necessarily an element of Δ because it is unlikely that all jobs have maximum duration at the same time. The set Δ is a *box uncertainty set* if $\Delta = \Pi_{i \in J}[\delta_i^-, \delta_i^+]$ for some $\delta^-, \delta^+ \in \mathbb{R}_+^J$. The set Δ is a *budgeted uncertainty set* associated with vector $\hat{\delta} \in \mathbb{R}_+^J$ and integer $\Gamma \in \{1, \dots, n\}$ if $\Delta = \left\{ (\hat{\delta}_i u_i)_{i \in J} \mid u_i \in \{0, 1\}^J, \sum_{i \in J} u_i \leq \Gamma \right\}$. Budget Γ corresponds to the maximum number of disruptions that can occur at the same time (Bertsimas and Sim (2004)). In practice budgeted uncertainty sets may seem unrealistic because durations of jobs can take only the two values p_i and $p_i + \hat{\delta}_i$. The set Δ is a *relaxed budgeted uncertainty set* associated with vector $\hat{\delta} \in \mathbb{R}_+^J$ and integer $\Gamma \in \{1, \dots, n\}$ if $\Delta = \left\{ (\hat{\delta}_i u_i)_{i \in J} \mid u_i \in [0, 1]^J, \sum_{i \in J} u_i \leq \Gamma \right\}$. In the latter case the budget can be dispatched over a larger number of jobs, compared to that in the budgeted uncertainty case.

2.1 A General-Purpose Graph Model

Let Δ be a general uncertainty set. For every $i, j \in \bar{J}$, let $L_{ij}^\Delta = \max_{\delta \in \Delta} L_{G(p+\delta)}(i, j)$. Let G^Δ be the weighted digraph obtained by taking the transitive closure of the precedence graph G , and adding the arc-weight L_{ij}^Δ on each arc (i, j) with $j \neq t$, and the arc-weight $L_{G(p)}(i, t)$ on

each arc (i, t) . Given $H \subseteq J$, let $G^\Delta[H]$ denote the subgraph of G^Δ induced by $H \cup \{s, t\}$. An example of an instance with 5 jobs and processing times $p = (2, 1, 2, 1, 2)$ is represented in Figure 1(a) with a precedence graph $G(p)$ and in Figure 1(b) with the associated graph G^Δ for the box uncertainty set $\Delta = \Pi_{i \in J}[0, \delta_i^+]$ with $\delta^+ = (2, 1, 2, 1, 1)$.

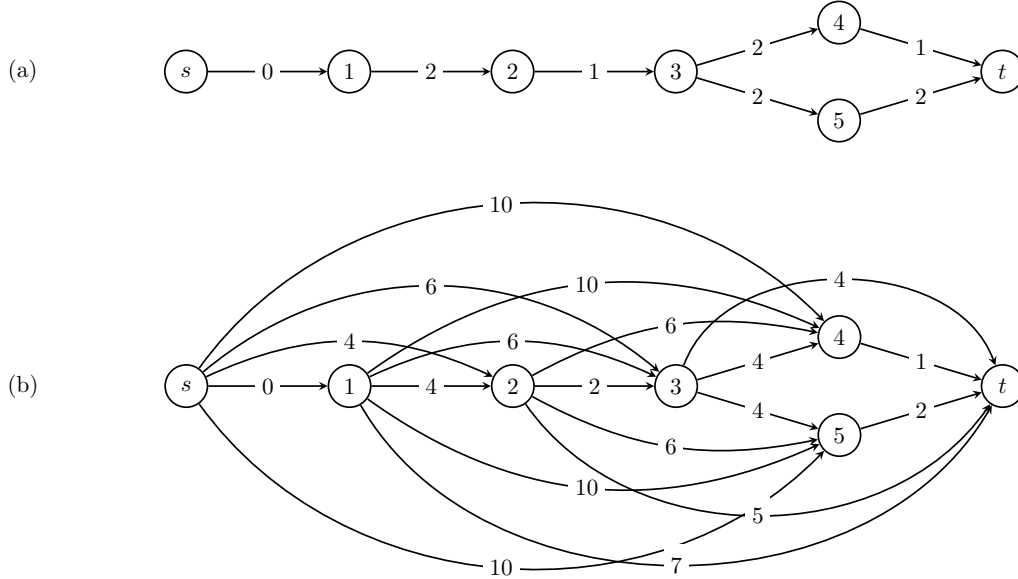


Figure 1: (a) A precedence graph $G(p)$ (b) The associated graph G^Δ for $\Delta = \Pi_{i \in J}[0, \delta_i^+]$ with $\delta^+ = (2, 1, 2, 1, 1)$.

We prove the following result.

Theorem 1. *Let $H \subseteq J$.*

(a) *Let x be a schedule of $G(p)$. The set H is anchored w.r.t. schedule x if and only if $x_j - x_i \geq L_{ij}^\Delta$ for every $i, j \in H \cup \{s\}$.*

(b) *The set H is anchored w.r.t. some baseline schedule x^0 if and only if $L_{G^\Delta[H]}(s, t) \leq M$.*

Proof. Note first that equivalence (a) follows from Lemma 1. Indeed, the set H is anchored w.r.t. x if and only if for every $\delta \in \Delta$, we have $x_j - x_i \geq L_{G(p+\delta)}(i, j)$ for every $i, j \in H \cup \{s\}$, which is equivalent to $x_j - x_i \geq \max_{\delta \in \Delta} L_{G(p+\delta)}(i, j) = L_{ij}^\Delta$.

Let us now prove equivalence (b). Assume H is anchored w.r.t. a baseline schedule x^0 . As a consequence of (a), it holds that $x_j^0 - x_i^0 \geq L_{ij}^\Delta$ for every $i, j \in H \cup \{s\}$. Also $x_t^0 - x_s^0 \geq L_{G(p)}(s, t)$ since x^0 is a schedule of $G(p)$. Hence $x_{H \cup \{s, t\}}^0$ is a schedule of $G^\Delta[H]$ and $x_t^0 \leq M$, thus $L_{G^\Delta[H]}(s, t) \leq M$.

Conversely, assume $L_{G^\Delta[H]}(s, t) \leq M$. Let \tilde{G} be the (multi)graph obtained from $G(p)$ by adding all the arcs of $G^\Delta[H]$. The claim is that $L_{\tilde{G}}(s, t) \leq M$. Let P be an $s-t$ path in \tilde{G} .

Let h_1, \dots, h_q be the vertices of $H \cap P$, indexed in the order they appear in the path. Let $h_0 = s$ and $h_{q+1} = t$ for the simplicity of notation. Let P_k be the subpath of P from h_k to h_{k+1} . Then for $k < q$, $\ell_{\tilde{G}}(P_k) \leq L_{h_k h_{k+1}}^\Delta$: indeed either P_k is the arc $(h_k h_{k+1})$ from $G^\Delta[H]$ of length $L_{h_k h_{k+1}}^\Delta$, or P_k is a path in $G(p)$, hence since $\Delta \subseteq \mathbb{R}_+^J$, an upper bound on the length of P_k is $L_{h_k h_{k+1}}^\Delta$. Also $\ell_{\tilde{G}}(P_q) \leq L_{G(p)}(h_q, t)$. Therefore, the length of P in \tilde{G} is at most the length of the path (s, h_1, \dots, h_q, t) in $G^\Delta[H]$. It follows that $L_{\tilde{G}}(s, t) \leq L_{G^\Delta[H]}(s, t) \leq M$. As a consequence from the claim, there exists a schedule x^0 of \tilde{G} such that $x_t^0 \leq M$, hence x^0 is a baseline schedule. Also $x_j^0 - x_i^0 \geq L_{ij}^\Delta$ for every $i, j \in H \cup \{s\}$ by definition of \tilde{G} , which implies with (a) that H is anchored w.r.t. x^0 . \square

Referring to the example shown in Figure 1, consider the subset of jobs $H = \{1, 5\}$. The longest s - t path in $G^\Delta[H]$ is $(s, 1, 5, t)$ and has length 12. From Theorem 1, for the considered uncertainty set Δ the subset $H = \{1, 5\}$ can then be anchored w.r.t. a baseline schedule if and only if the deadline M is at least 12.

Note that the graph G^Δ is dense since it is the transitive closure of the precedence graph. Moreover, the computation of the arc-weights L_{ij}^Δ might already be a hard problem, depending on the definition of Δ . Indeed computing a weight L_{ij}^Δ amounts to solving a maximization problem on Δ . More details are provided in the complexity analysis presented in Section 3.1.

2.2 A Layered Graph for Budgeted Uncertainty

Let $\hat{\delta} \in \mathbb{R}_+^J$, $\Gamma \in \{1, \dots, n\}$ and let Δ be the budgeted uncertainty set $\Delta = \{(\hat{\delta}_i u_i)_{i \in J} \mid u_i \in \{0, 1\}^J, \sum_{i \in J} u_i \leq \Gamma\}$. The objective of this section is to provide a dedicated graph model for budgeted uncertainty.

The *layered graph* G^{lay} is defined as follows. It contains $\Gamma + 1$ copies of the precedence graph $G(p)$, called *layers*, indexed by $\gamma \in \{0, \dots, \Gamma\}$. The layer Γ is called *upper layer*, the layer 0 is the *lowest layer*. The vertices in layer γ are denoted by $s^\gamma, 1^\gamma, \dots, n^\gamma, t^\gamma$. The arcs of each copy of $G(p)$ are called *horizontal arcs*. Additionally, for each arc (i, j) of $G(p)$, for each $\gamma \in \{0, \dots, \Gamma - 1\}$, the layered graph contains a *transversal arc* $(i^{\gamma+1}, j^\gamma)$ with weight $p_i + \hat{\delta}_i$. Given $H \subseteq J$, the *layered graph* $G^{\text{lay}}(H)$ associated with H is defined from the layered graph G^{lay} by adding, for each job $i \in H$, for each $\gamma \in \{0, \dots, \Gamma - 1\}$, a *vertical arc* (i^γ, i^Γ) with weight 0.

Referring to the instance presented in Figure 1, the corresponding layered graph $G^{\text{lay}}(H)$ for $H = \{1, 5\}$, $\hat{\delta} = \delta^+ = (2, 1, 2, 1, 1)$ and $\Gamma = 2$ is represented in Figure 2, with copies of jobs in H appearing in gray.

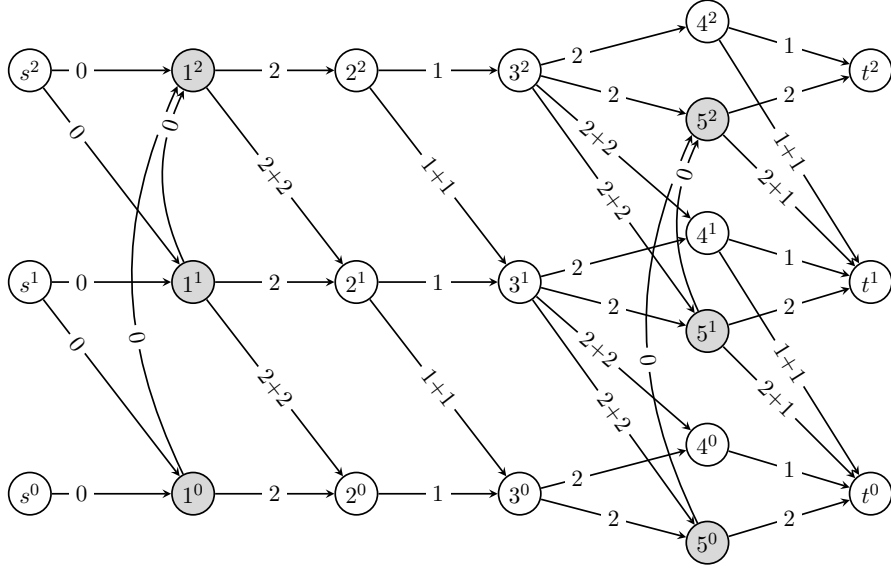


Figure 2: The layered graph $G^{\text{lay}}(H)$ associated with precedence graph from Figure 1, for $H = \{1, 5\}$ and $\Gamma = 2$.

A key property of the layered graph is stated in Lemma 2.

Lemma 2. *Let $i, j \in \bar{J}$. In the layered graph G^{lay} , the longest path from i^Γ to a copy of j has length L_{ij}^Δ .*

Proof. First, let us prove that by construction of the layered graph, the length of any path P from i^Γ to a copy of j is upper-bounded by L_{ij}^Δ . For every job k on the path P , let δ_k^* be equal to $\widehat{\delta}_k$ if P goes through a transversal arc outgoing from a copy of k , and $\delta_k^* = 0$ otherwise. Then the length of P in the layered graph is equal to the length of the corresponding path in $G(p + \delta^*)$. Also, path P uses at most Γ transversal arcs, hence $\delta^* \in \Delta$ and the length of P is at most L_{ij}^Δ . Let us now show that the equality holds by exhibiting for every i, j a path P of the layered graph of length L_{ij}^Δ . The value L_{ij}^Δ is equal to $\ell_{G(p+\delta^*)}(Q)$ for some $\delta^* \in \Delta$ and some path Q in the precedence graph. Let P be the path in the layered graph obtained by starting at vertex i^Γ and following the arcs of Q : path P follows a horizontal arc if the corresponding job is not disrupted in δ^* , or a transversal arc if the corresponding job is disrupted in δ^* . Then the length of P is equal to L_{ij}^Δ . \square

We note that the layered graph G^{lay} can be interpreted as the state graph corresponding to the dynamic programming algorithm from (Minoux (2007b)).

Theorem 2. *Let Δ be a budgeted uncertainty set. Let $H \subseteq J$.*

(a) *Let z be a schedule of $G(p)$. The set H is anchored w.r.t. z if and only if there exists x a*

schedule of $G^{\text{lay}}(H)$ such that $x_i^\Gamma = z_i$ for every $i \in \bar{J}$.

(b) A set H is anchored w.r.t. some baseline schedule if and only if $L_{G^{\text{lay}}(H)}(s^\Gamma, t^\Gamma) \leq M$.

Proof. (a) Assume the set H is anchored w.r.t. z , or equivalently with Theorem 1, that $z_j - z_i \geq L_{ij}^\Delta$ for every $i, j \in H \cup \{s\}$. Let us prove that $z_j - z_i \geq L_{G^{\text{lay}}(H)}(i^\Gamma, j^\Gamma)$ for every $i, j \in \bar{J}$: indeed by Lemma 1 it is a sufficient condition for the existence of a schedule x of $G^{\text{lay}}(H)$ such that $x_i^\Gamma = z_i$ for every $i \in \bar{J}$. Let P be an $i^\Gamma - j^\Gamma$ path in $G^{\text{lay}}(H)$, and let h_1, \dots, h_q be the jobs of H corresponding to vertical arcs of P . Let $h_0 = i$ and $h_{q+1} = j$ for the ease of notation. Consider $P_k, k \in \{0, \dots, q\}$ the subpath of P going from h_k^Γ to a copy of h_{k+1} . By definition of the h_k 's the path P_k uses no vertical arc: it is a path in G^{lay} . By Lemma 2, it comes $\ell_{G^{\text{lay}}(H)}(P_k) \leq L_{h_k h_{k+1}}^\Delta$. By assumption on z it holds that $L_{h_k h_{k+1}}^\Delta \leq z_{h_{k+1}} - z_{h_k}$, hence $\ell_{G^{\text{lay}}(H)}(P_k) \leq z_{h_{k+1}} - z_{h_k}$. Summing up the inequality over $k \in \{0, \dots, q\}$ we obtain $\ell_{G^{\text{lay}}(H)}(P) \leq z_{h_{q+1}} - z_{h_0} = z_j - z_i$. This holds for every $i^\Gamma - j^\Gamma$ path P , therefore $z_j - z_i \geq L_{G^{\text{lay}}(H)}(i^\Gamma, j^\Gamma)$.

Conversely, assume the existence of such a schedule x . Let us prove that $z_j - z_i \geq L_{ij}^\Delta$ for every $i, j \in H \cup \{s\}$. Let $i, j \in H \cup \{s\}$, and consider $\gamma \in \{0, \dots, \Gamma\}$ such that there is a path from i^Γ to j^γ which is a longest path between i^Γ and a copy of j in G^{lay} . Then $z_j - z_i = x_j^\Gamma - x_i^\Gamma = x_j^\Gamma - x_j^\gamma + x_j^\gamma - x_i^\Gamma$, where $x_j^\Gamma - x_j^\gamma \geq 0$ because of the vertical arc (j^γ, j^Γ) , and $x_j^\gamma - x_i^\Gamma \geq L_{ij}^\Delta$ by definition of γ and with Lemma 2. Thus $z_j - z_i \geq L_{ij}^\Delta$ and with Theorem 1, the set H is anchored w.r.t. z .

Equivalence (b) can then be proven as follows. From (a), a subset H is anchored w.r.t. a baseline schedule if and only if there exists a schedule of $G^{\text{lay}}(H)$ whose restriction to the upper layer has makespan at most M . The existence of such a schedule is equivalent to $L_{G^{\text{lay}}(H)}(s^\Gamma, t^\Gamma) \leq M$. \square

Referring to the layered graph represented in Figure 2, the longest $s^2 - t^2$ path is $(s^2, 1^2, 2^1, 3^1, 5^0, 5^2, t^2)$, with length 11. From Theorem 2, for uncertainty budget $\Gamma = 2$, the subset $H = \{1, 5\}$ can be anchored w.r.t. a baseline schedule if and only if the deadline M is at least 11.

Let us now derive a compact mixed integer programming (MIP) formulation for AnchRobPSP under budgeted uncertainty, using the layered graph. A real variable $x_j^\gamma \geq 0$ is associated with every vertex j^γ of the layered graph; a binary variable $h_j \in \{0, 1\}$ is associated with every job $j \in J$, so that $h_j = 1$ if j is anchored, and $h_j = 0$ otherwise. Let \mathcal{A} denote the arc-set of the precedence graph G . Let $D_j = L_{G(p+\hat{\delta})}(s, j) - L_{G(p)}(s, j)$ for every $j \in J$. Consider the following program.

$$\begin{aligned}
(P) \quad & \max \sum_{i \in J} a_i h_i \\
\text{s.t.} \quad & x_j^\gamma - x_i^\gamma \geq p_i && \forall (i, j) \in \mathcal{A}, \forall \gamma \in \{0, \dots, \Gamma\} && (1) \\
& x_j^\gamma - x_i^{\gamma+1} \geq p_i + \widehat{\delta}_i && \forall (i, j) \in \mathcal{A}, \forall \gamma \in \{0, \dots, \Gamma - 1\} && (2) \\
& x_j^\Gamma - x_j^\gamma \geq -D_j(1 - h_j) && \forall j \in J, \forall \gamma \in \{0, \dots, \Gamma - 1\} && (3) \\
& x_t^\Gamma \leq M && && (4) \\
& x_j^\gamma \geq 0 && \forall j \in \bar{J}, \forall \gamma \in \{0, \dots, \Gamma\} \\
& h_j \in \{0, 1\} && \forall j \in J
\end{aligned}$$

Proposition 1. *Program (P) is a valid MIP formulation for AnchRobPSP under uncertainty budget Γ .*

Proof. Proof. With Theorem 2, AnchRobPSP is equivalent to maximizing $a(H)$ subject to $H \subseteq J$, x schedule of $G^{\text{lay}}(H)$, $x_t^\Gamma \leq M$. Let us show that there exists a feasible pair (x, H) of the problem with objective value $a(H)$ if and only if there exists a feasible solution of program (P) with objective value equal to $a(H)$. First note that, given a feasible solution (x, h) of the MIP, it holds that x is a schedule of $G^{\text{lay}}(H)$ where $H = \{i \in J \mid h_i = 1\}$. Indeed constraints (1) and (2) correspond to horizontal and transversal arcs respectively, and constraints (3) correspond to vertical arcs for jobs $i \in H$. Conversely, consider $H \subseteq J$ such that there exists a schedule x of $G^{\text{lay}}(H)$, $x_t^\Gamma \leq M$. Let x' be the earliest schedule of $G^{\text{lay}}(H)$. Note that $x_t^\Gamma \leq M$. Let h be the indicator vector of H . Then (x', h) is a feasible solution of the MIP with objective value $a(H)$. Indeed x' clearly satisfies constraints (1)-(2)-(4), and constraint (3) when $h_j = 1$. It remains to prove that x' also satisfies constraint (3) when $h_j = 0$. We have $x_j^\gamma = L_{G^{\text{lay}}(H)}(s^\Gamma, j^\gamma) \leq L_{G^{\text{lay}}(J)}(s^\Gamma, j^\gamma) \leq L_{G(p+\widehat{\delta})}(s, j)$. Indeed in $G^{\text{lay}}(J)$ the longest $s^\Gamma - j^\gamma$ path uses alternatively vertical and transversal arcs, hence its length is the length of the corresponding $s - j$ path in $G(p + \widehat{\delta})$. Also $x_j^\Gamma \geq L_{G(p)}(s, j)$, hence $x_j^\Gamma - x_j^\gamma \leq L_{G(p+\widehat{\delta})}(s, j) - L_{G(p)}(s, j) = D_j$. \square

Note that the number of constraints and variables in (P) is polynomial in n ; also the values D_j can be computed in polynomial time in a preprocessing step.

2.3 Convex Hull of the Uncertainty Set

In robust problems, it is frequently the case that the uncertainty set can be convexified without changing feasible solutions of the problem (see, e.g., Remark 1 in (Pessoa et al. (2015))). A consequence of the general-purpose graph model and Theorem 1 is that a similar result holds for AnchRobPSP.

Proposition 2. *Let Δ be an uncertainty set and $\text{conv}(\Delta)$ be the convex hull of Δ . The instances $(G(p), \Delta, a, M)$ and $(G(p), \text{conv}(\Delta), a, M)$ of *AnchRobPSP* have the same feasible solutions.*

Proof. Note that since $\Delta \subseteq \text{conv}(\Delta)$, all solutions for uncertainty set $\text{conv}(\Delta)$ are feasible for uncertainty set Δ , by definition of anchored sets. We now show the converse using the graphs G^Δ and $G^{\text{conv}(\Delta)}$. The main claim is that $L_{ij}^\Delta = L_{ij}^{\text{conv}(\Delta)}$ for any pair i, j of jobs. Indeed $L_{ij}^\Delta = \max_{\delta \in \Delta} \max_{P \in \mathcal{P}_{ij}} \ell_{G(p+\delta)}(P) = \max_{P \in \mathcal{P}_{ij}} \max_{\delta \in \Delta} \ell_{G(p+\delta)}(P)$ where \mathcal{P}_{ij} is the set of i - j paths of $G(p)$. Given $P \in \mathcal{P}_{ij}$, the quantity $\max_{\delta \in \Delta} \ell_{G(p+\delta)}(P)$ is the maximum of the linear function $\sum_{k \in P} (p_k + \delta_k)$ for $\delta \in \Delta$. Hence it equals the maximum over $\text{conv}(\Delta)$, that is, $\max_{\delta \in \text{conv}(\Delta)} \ell_{G(p+\delta)}(P) = \max_{\delta \in \Delta} \ell_{G(p+\delta)}(P)$. Taking the maximum over \mathcal{P}_{ij} , we obtain $L_{ij}^\Delta = L_{ij}^{\text{conv}(\Delta)}$. Consequently, the graphs G^Δ and $G^{\text{conv}(\Delta)}$ are the same. From Theorem 1 it follows that it is equivalent to solve *AnchRobPSP* on any of the two uncertainty sets. \square

Let us consider the budgeted uncertainty set associated with $\widehat{\delta} \in \mathbb{R}_+^J$ and budget $\Gamma \in \{1, \dots, n\}$. Proposition 2 implies the following corollary.

Corollary 1. **AnchRobPSP* for budgeted uncertainty set $\{(\widehat{\delta}_i u_i)_{i \in J} \mid u_i \in \{0, 1\}^J, \sum_{i \in J} u_i \leq \Gamma\}$ and *AnchRobPSP* for relaxed budgeted uncertainty set $\{(\widehat{\delta}_i u_i)_{i \in J} \mid u_i \in [0, 1]^J, \sum_{i \in J} u_i \leq \Gamma\}$ have the same feasible solutions.*

Consequently, the layered graph from Section 2.2 can also be used for the relaxed budgeted uncertainty set $\{(\widehat{\delta}_i u_i)_{i \in J} \mid u_i \in [0, 1]^J, \sum_{i \in J} u_i \leq \Gamma\}$.

3 Complexity of *AnchRobPSP*

To study the complexity of *AnchRobPSP*, we consider two distinct complexity issues, depending on whether the inputs of the problem contain $G(p)$ and Δ , or directly the graph G^Δ with the arc-weights L_{ij}^Δ . On the one hand, if the values L_{ij}^Δ are part of the input, the decision version of *AnchRobPSP* is in NP, since checking whether $L_{G^\Delta[H]}(s, t) \leq M$ can be done in polynomial time. On the other hand, if the values L_{ij}^Δ are not part of the input, we show that for some uncertainty sets it is co-NP-hard to decide if a set H is anchored, and thus *AnchRobPSP* is co-NP-hard. Let us start with this latter point.

3.1 *AnchRobPSP* is at least as hard as computing L_{ij}^Δ

Let \mathcal{D} be a family of uncertainty sets. Let *WORST-CASE LONGEST PATH WLP*(\mathcal{D}) be the following problem: given a precedence graph $G(p)$, a job $j \in J$, $\Delta \in \mathcal{D}$, $L \geq 0$, is it true that for every $\delta \in \Delta$, the longest s - j path in $G(p + \delta)$ has length at most L ? Note that this problem is to decide whether $L_{sj}^\Delta \leq L$. Let also *ANCH*(\mathcal{D}) be the decision version of

AnchRobPSP over the family of uncertainty sets \mathcal{D} , that is: given a precedence graph $G(p)$, $\Delta \in \mathcal{D}$, $M \geq 0$, $a \in \mathbb{R}_+^J$, $A \geq 0$, is there $H \subseteq J$ such that $a(H) \geq A$ and $L_{G\Delta[H]}(s, t) \leq M$?

Lemma 3. *There exists a polynomial reduction from WLP(\mathcal{D}) to ANCH(\mathcal{D}).*

Proof. Let $\mathcal{I}_{WLP} = (G(p), j, \Delta, L)$ be an instance of WLP(\mathcal{D}). An associated instance $\mathcal{I}_A = (G'(p'), \Delta', M', a', A')$ of ANCH(\mathcal{D}) problem is built as follows. Let $G' = G$, and $p'_i = p_i$ if job i is a predecessor of j , and $p'_i = 0$ otherwise (including $p'_j = 0$). Let $M' = L$ and $\Delta' = \Delta$. Finally let $a'_i = 1$ for every $i \neq d$, and $a'_d = A' = n$, where n is the number of jobs in $G(p)$.

For instance \mathcal{I}_A , the anchoring weights are such that a subset H of jobs satisfies $a'(H) \geq A'$ if and only if it contains job j . If \mathcal{I}_{WLP} has a ‘yes’ answer, i.e., if $L_{sj}^\Delta \leq L$, then $L_{G'\Delta'[\{j\}]}(s, t) = L_{sj}^\Delta + L_{G'(p')}(j, t) \leq L = M'$. Thus the singleton $\{j\}$ yields a ‘yes’ answer for \mathcal{I}_A . Conversely, if H is anchored and $a'(H) \geq A'$, then $j \in H$. The value L_{sj}^Δ is the arc-weight of the arc (s, j) of $G'\Delta'[H]$, hence $L_{sj}^\Delta \leq L_{G'\Delta'[H]}(s, t) \leq M' = L$. \square

Note the reduction can be modified so that anchoring weights are all equal to one in \mathcal{I}_A . Indeed it suffices to replace job d by n duplicates of d , all with the same predecessors and successors as d . The reduction then applies similarly.

Let us fix notation for the KNAPSACK problem in order to show the following theorem. The KNAPSACK problem is: given a set I of m items, $w \in \mathbb{N}^I$, $v \in \mathbb{N}^I$, $W \in \mathbb{N}$, $V \in \mathbb{N}$, is there a subset $S \subseteq I$ of items such that $w(S) \leq W$ and $v(S) \geq V$.

Theorem 3. *AnchRobPSP is co-NP-hard for upper-bounded uncertainty sets, even with unit anchoring weights.*

Proof. Proof. We claim that the complement of WLP(\mathcal{D}) is NP-hard when \mathcal{D} denotes the family of sets Δ defined by $\Delta = \left\{ (\widehat{\delta}_i u_i)_{i \in J} \mid u_i \in \{0, 1\}^J, \sum_{i \in J} \alpha_i u_i \leq \beta \right\}$ for some $\alpha \in \mathbb{N}^J$ and $\beta \in \mathbb{N}$. Every Δ of that form is an upper-bounded uncertainty set. From the reduction of Lemma 3, the claim will imply that AnchRobPSP is co-NP-hard in this case. Let us now show this claim. The complement of WLP(\mathcal{D}) is: given $G(p)$, job j , set Δ defined by $\widehat{\delta}$, α , β , and $L \geq 0$, is there $u \in \{0, 1\}^J$ such that $\sum_{i \in J} \alpha_i u_i \leq \beta$ and $L_{G(p+\widehat{\delta}u)}(s, j) > L$? The proof is a reduction from KNAPSACK. Let $\mathcal{I}_{KP} = (I = \{1, \dots, m\}, w_i, v_i, W, V)$ be an instance of the KNAPSACK problem. The associated instance $\mathcal{I}_{WLP} = (G(p), j, \widehat{\delta}, \alpha, \beta, L)$ of WLP(\mathcal{D}) is defined as follows. Let $L = V - 1$, let $G(p)$ be a path $(s, 1, \dots, m, j, t)$ with arc-weights $p_i = 0$ for every $i \in I$, $p_j = 0$. Let $\widehat{\delta}_i = v_i$, $\alpha_i = w_i$ for every $i \in I$, and $\beta = W$. Given $u \in \{0, 1\}^J$, the constraint $\sum_{i \in J} \alpha_i u_i \leq \beta$ then writes $\sum_{i \in I} w_i u_i \leq W$, and $L_{G(p+\widehat{\delta}u)}(s, j) = \sum_{i \in I} (p_i + \widehat{\delta}_i u_i) = \sum_{i \in I} v_i u_i$. Hence there exists $u \in \{0, 1\}^J$ that yields a ‘yes’ answer for the complement of WLP(\mathcal{D}) on \mathcal{I}_{WLP} if and only if it yields a ‘yes’ answer for the instance \mathcal{I}_{KP} of the KNAPSACK problem. \square

3.2 NP-Completeness for Budgeted Uncertainty

In this section we show that for \mathcal{D}^Γ the family of budgeted uncertainty sets, the decision version $\text{ANCH}(\mathcal{D}^\Gamma)$ of AnchRobPSP is NP-complete. First note that for every $\Delta \in \mathcal{D}^\Gamma$ the values L_{ij}^Δ can be computed in polynomial time, e.g., by the dynamic programming algorithm from (Minoux (2007b)). Hence $\text{ANCH}(\mathcal{D}^\Gamma)$ is in NP. $\text{ANCH}(\mathcal{D}^\Gamma)$ is proven NP-complete in the restricted case where the precedence graph is a path. We establish the following lemma.

Lemma 4. *Let G be an s - t path $(s, 1, 2, \dots, n, t)$. A subset $\{h_1, \dots, h_q\} \subseteq J$ (with $h_1 < \dots < h_q$) is anchored for a baseline schedule if and only if $L_{sh_1}^\Delta + L_{h_1h_2}^\Delta + \dots + L_{h_{q-1}h_q}^\Delta + L_{G(p)}(h_q, t) \leq M$.*

Proof. We show that $L_{ij}^\Delta \leq L_{ik}^\Delta + L_{kj}^\Delta$ for every $i \leq k \leq j$. Indeed if P_{ij} denotes the (unique) i - j path in G , for some $\delta^* \in \Delta$ it holds that $L_{ij}^\Delta = \ell_{G(p+\delta^*)}(P_{ij}) = \ell_{G(p+\delta^*)}(P_{ik}) + \ell_{G(p+\delta^*)}(P_{kj}) \leq L_{ik}^\Delta + L_{kj}^\Delta$. This inequality implies that the longest s - t path in $G^\Delta[H]$ is the path $(s = h_0, h_1, \dots, h_q, t = h_{q+1})$ going through all jobs of H . The result follows from Theorem 1. \square

Theorem 4. *$\text{ANCH}(\mathcal{D}^\Gamma)$ is NP-complete, even if the precedence graph is a path and $\Gamma = 1$.*

Proof. The proof is a reduction from KNAPSACK . The problem $\text{ANCH}(\mathcal{D}^\Gamma)$ on a path with $\Gamma = 1$ is stated as follows: given a precedence graph $G(p)$ which is an s - t path, $\hat{\delta} \in \mathbb{R}_+^J$, $M \geq 0$, $a \in \mathbb{R}_+^J$, $A \geq 0$, is there a subset $H \subseteq J$ such that $a(H) \geq A$ and $L_{G^\Delta[H]}(s, t) \leq M$, where Δ is the budgeted uncertainty set defined by $\hat{\delta}$ and $\Gamma = 1$. Let $\mathcal{I}_{KP} = (I = \{1, \dots, m\}, w_i, v_i, W, V)$ be an instance of the KNAPSACK problem. The associated instance $\mathcal{I}_A = (J, p_i, \hat{\delta}, M, a_i, A)$ of $\text{ANCH}(\mathcal{D}^\Gamma)$ is as follows:

- the set of jobs is $J = \{0, 1, \dots, m, m+1\}$, where we assume w.l.o.g. that the items $1, \dots, m$ are sorted by decreasing weight: $w_1 \geq \dots \geq w_m$
- the precedence graph is the path $(0, 1, \dots, m+1)$ and $p_i = 0$ for every $i \in J$
- $\hat{\delta}_i = w_i$ for every $i \in I$, $\hat{\delta}_0 = 1 + \max_{i \in I} w_i$, $\hat{\delta}_{m+1} = 0$ (hence $\hat{\delta}_0 \geq \hat{\delta}_1 \geq \dots \geq \hat{\delta}_{m+1}$)
- $M = W + \hat{\delta}_0$
- $a_i = v_i$ for every $i \in I$, $a_0 = 0$, $a_{m+1} = \bar{v}$ where $\bar{v} = 1 + \sum_{i \in I} v_i$
- $A = V + \bar{v}$

The size of \mathcal{I}_A is polynomial in the size of \mathcal{I}_{KP} . We prove three claims on instance \mathcal{I}_A .

(i) *Let $H \subseteq J$ and $S = H \cap I$. Then $a(H) \geq A \iff v(S) \geq V$ and $m+1 \in H$.* Indeed $a(H) \geq A$ writes $v(S) + \bar{v} \mathbf{1}_{m+1 \in H} \geq V + \bar{v}$. If $m+1 \notin H$ then the inequality cannot be satisfied since $\bar{v} > v(S)$.

(ii) *For every $i, j \in J$ such that $i < j$, $L_{ij}^\Delta = \hat{\delta}_i$ and $L_{si}^\Delta = \hat{\delta}_0$.* Indeed by definition $L_{ij}^\Delta = \max_{\delta \in \Delta} \sum_{i \leq k < j} \delta_k = \max_{i \leq k < j} \hat{\delta}_k$ which equals $\hat{\delta}_i$ if $i \neq s$ and $\hat{\delta}_0$ otherwise.

(iii) Let $H \subseteq J$ such that $m + 1 \in H$ and let $S = H \cap I$. Then $L_{G^\Delta[H]}(s, t) \leq M \iff w(S) \leq W$. Let $H = \{h_1, \dots, h_q\}$ with $h_1 < \dots < h_q$. By Lemma 4 and claim (ii), $L_{G^\Delta[H]}(s, t) = \widehat{\delta}_0 + \widehat{\delta}_{h_1} + \dots + \widehat{\delta}_{h_{q-1}}$. Note that $S = \{h_1, \dots, h_{q-1}\}$ since $m + 1 \in H$. Thus $L_{G^\Delta[H]}(s, t) \leq M \iff \widehat{\delta}_0 + \widehat{\delta}(S) \leq M \iff w(S) \leq W$.

Let us now prove that the instance \mathcal{I}_{KP} of the KNAPSACK problem has a ‘yes’ answer if and only if the instance \mathcal{I}_A of ANCH(\mathcal{D}^Γ) has a ‘yes’ answer. If there exists some $S \subseteq I$ such that $w(S) \leq W$ and $v(S) \geq V$, then let $H = S \cup \{m + 1\}$: by (i) it comes $a(H) \geq A$, and by (iii) $L_{G^\Delta[H]}(s, t) \leq M$. Conversely assume there exists $H \subseteq J$ such that $a(H) \geq A$ and $L_{G^\Delta[H]}(s, t) \leq M$, and let $S = H \cap I$. Then by (i) $v(S) \geq V$ and $m + 1 \in H$, and by (iii) $w(S) \leq W$. Hence the problem ANCH(\mathcal{D}^Γ) is NP-complete, even if the precedence graph is a path and $\Gamma = 1$. \square

Using Corollary 1, a consequence of Theorem 4 is the following.

Corollary 1. *AnchRobPSP is NP-hard under relaxed budgeted uncertainty.*

3.3 Discussion on complexity aspects for given L_{ij}^Δ

We finally discuss some aspects of the complexity of the problem, when the values L_{ij}^Δ are part of the input of the problem. From Theorem 1, the problem AnchRobPSP is equivalent to finding a subset $H \subseteq J$ such that $L_{G^\Delta[H]}(s, t) \leq M$, and $a(H)$ is maximized. Equivalently, by considering the set $I = J \setminus H$ of non-anchored jobs, the problem is to find $I \subseteq J$ such that $L_{G^\Delta[J \setminus I]}(s, t) \leq M$, and $a(I)$ is minimized. This can be recognized as the min Weight Vertex Blocker to Longest Path Problem (WVBLP), which is a variant of network interdiction problems studied in the literature (Israeli and Wood (2002), Boros et al. (2005)). Given an instance of a maximization problem on a network, vertex blocker problems look for a subset of vertices to remove, so that the optimum of the maximization problem in the remaining graph falls below a given threshold. Hence AnchRobPSP corresponds to WVBLP on G^Δ , thus raising the question of the complexity of the WVBLP on G^Δ graphs.

Many vertex blocker problems have been shown to be NP-complete (Boros et al. (2005)), but these results cannot be directly applied to AnchRobPSP, since it is the restriction of WVBLP to very structured instances, namely the G^Δ graphs. Recall that the G^Δ are transitive closures of acyclic graphs, and their arc-weights are values L_{ij}^Δ . Theorem 4 provides an NP-hard case of the problem WVBLP on G^Δ instances. Note that the reduction involves non-unit anchoring weights. The complexity of AnchRobPSP when the values L_{ij}^Δ are part of the input, and with unit anchoring weights, is still an open question.

4 Algorithms for special cases of AnchRobPSP

In this section we provide an algorithm (Algorithm \mathbf{A}^+) to compute a feasible solution for AnchRobPSP under upper-bounded uncertainty, and show that AnchRobPSP under box uncertainty is a polynomial case where Algorithm \mathbf{A}^+ is exact. For budgeted uncertainty, polynomial algorithms are exhibited for special cases, and a dynamic programming approach is proposed for series-parallel precedence graphs.

4.1 Algorithm \mathbf{A}^+ for upper-bounded uncertainty

In this section the set Δ is an upper-bounded uncertainty set, i.e., there exists $\delta^+ \in \mathbb{R}_+^J$ such that $\delta_i^+ \geq \delta_i$ for every $i \in J$ for every $\delta \in \Delta$. Let \underline{x} denote the earliest schedule of $G(p + \delta^+)$. A first remark is that if the makespan of \underline{x} is at most M , then (\underline{x}, J) is an optimal solution of AnchRobPSP. Indeed \underline{x} is then both a baseline schedule and a feasible second-stage schedule for every $\delta \in \Delta$, since δ^+ is an upper bound on Δ . However, in general the makespan of \underline{x} is larger than M and then not all jobs can be anchored. Let \bar{x} denote the latest schedule of $G(p)$ such that $\bar{x}_i = M$. Consider the following algorithm.

Algorithm \mathbf{A}^+

Input: $G(p)$, δ^+ , M .

Output: a solution (x, H) of AnchRobPSP.

Compute $\underline{x}_i = L_{G(p+\delta^+)}(s, i)$ for every $i \in \bar{J}$;

Compute $\bar{x}_i = M - L_{G(p)}(i, t)$ for every $i \in \bar{J}$;

Let $x_i = \min\{\underline{x}_i, \bar{x}_i\}$ for every $i \in \bar{J}$;

Let $H = \{i \in J \mid \underline{x}_i \leq \bar{x}_i\}$;

Return (x, H) .

Let us first prove that Algorithm \mathbf{A}^+ provides a feasible solution for AnchRobPSP for any upper-bounded uncertainty set.

Proposition 3. *If Δ is upper-bounded by δ^+ , Algorithm \mathbf{A}^+ returns a feasible solution of AnchRobPSP.*

Proof. Let (x, H) be the output of Algorithm \mathbf{A}^+ . First we show that x is a baseline schedule. It is clear that it has makespan at most M since $x_t \leq \bar{x}_t \leq M$. Also \bar{x} and \underline{x} are both schedules of $G(p)$, since $\delta^+ \geq 0$. Hence $x = \min\{\underline{x}, \bar{x}\}$ is also a schedule of $G(p)$. The set H is anchored with respect to x : indeed let us consider for every $\delta \in \Delta$ the same second-stage schedule \underline{x} . It is a schedule of $G(p + \delta)$ because δ^+ is an upper bound on Δ , and for every $i \in H$,

$\underline{x}_i = x_i$. It follows that the pair (x, H) is a feasible solution of the instance $(G(p), \Delta, a, M)$ of AnchRobPSP. \square

Note that a feasible solution of AnchRobPSP for a budgeted uncertainty set can thus be obtained by using Algorithm \mathbf{A}^+ with $\delta^+ = \widehat{\delta}$, since $\widehat{\delta}$ is an upper bound for Δ . It is also worth mentioning that solutions computed with Algorithm \mathbf{A}^+ have a specific structure. Namely, if H is the set returned by \mathbf{A}^+ and $i \notin H$, one can check that no job on the longest path from i to t in $G(p)$ belongs to H .

We now prove that box uncertainty is a polynomial case, where Algorithm \mathbf{A}^+ is exact.

Theorem 5. *For box uncertainty, Algorithm \mathbf{A}^+ solves AnchRobPSP in polynomial time.*

Proof. Let us show that \mathbf{A}^+ returns an optimal solution when Δ is the box $\Pi_{i \in J}[\delta_i^-, \delta_i^+]$. First it is feasible by Proposition 3. Let j be a job not anchored by the algorithm, i.e., such that $\underline{x}_j > \bar{x}_j$. Let x^0 be an arbitrary baseline schedule. Then $L_{G(p+\delta^+)}(s, j) + L_{G(p)}(j, t) = \underline{x}_j + M - \bar{x}_j > M$ by assumption on j , and since x^0 is a baseline schedule $M \geq x_t^0 - x_j^0 + x_j^0 - x_s^0 \geq L_{G(p)}(j, t) + x_j^0 - x_s^0$. It comes $L_{G(p+\delta^+)}(s, j) > x_j^0 - x_s^0$. From Lemma 1, there exists no schedule x^{δ^+} of $G(p+\delta^+)$ such that $x_j^{\delta^+} = x_j^0$. Since $\delta^+ \in \Delta$, job j cannot be in any anchored set. Since all anchoring weights are non-negative, the solution returned by the algorithm is optimal. \square

Note that the proof only requires that Δ is upper-bounded by δ^+ and that $\delta^+ \in \Delta$, which is more general than Δ being the box $\Pi_{i \in J}[\delta_i^-, \delta_i^+]$. The proof of Theorem 5 also yields as a corollary that the set H optimal for AnchRobPSP under box uncertainty is unique, and it is the set returned by Algorithm \mathbf{A}^+ .

4.2 Polynomial Cases for Budgeted Uncertainty

Under budgeted uncertainty, AnchRobPSP is NP-hard as shown in Theorem 4. In this section we exhibit polynomial special cases.

4.2.1 Path precedence graph and unit anchoring weights.

We first investigate the case where the precedence graph is a path. The reduction from Theorem 4 relies on the numerical values of anchoring weights. We show that in the case of unit anchoring weights the problem becomes solvable in polynomial time.

Theorem 6. *Let G be an $s-t$ path, and let Δ be a budgeted uncertainty set. If $a_i = 1$ for every $i \in J$, then AnchRobPSP is solvable in polynomial time.*

Proof. Proof. Under budgeted uncertainty, the arc-weights of graph G^Δ can be computed in polynomial time. Consider then the problem of finding an $s-t$ path in G^Δ with length

at most M , and with a maximum number of arcs. This problem is solvable in polynomial time as a polynomial case of the Resource-Constrained Longest Path Problem (Garey and Johnson (1979)). The claim is that this problem is equivalent to solving the special case of AnchRobPSP. Indeed from Lemma 4, there is a one-to-one correspondence between subsets of jobs and $s-t$ paths in G^Δ , and a subset H is feasible for AnchRobPSP if and only if the corresponding path has length at most M in G^Δ . Hence maximizing $a(H) = |H|$ amounts to maximizing the number of arcs in the associated $s-t$ path, under the constraint that the length of the path is at most M . \square

Note that the result holds not only for budgeted uncertainty, but also for any uncertainty set for which the arc-weights L_{ij}^Δ can be computed in polynomial time.

4.2.2 A special case with $p_i=0$, $\widehat{\delta}_i=1$, $\Gamma=1$, $a_i=1$.

Let the precedence graph G be any directed acyclic graph. Let us consider the special case U-AnchRobPSP where $p_i = 0$ and $a_i = 1$ for every $i \in J$ and Δ is a budgeted uncertainty with $\widehat{\delta}_i = 1$ for every $i \in J$ and budget $\Gamma = 1$. An instance of U-AnchRobPSP is thus formed with the graph G and the deadline M , assumed to be an integer. We will show that U-AnchRobPSP is a polynomial case, by an equivalence with a poset problem. Recall that a partial order \prec can be naturally defined from the precedence graph G by setting $i \prec j$ if there exists an $i-j$ path in G . Let MaxSubposet be the problem of finding, given a poset and an integer M , a max-size subposet in which all chains have size at most M .

Let (G, M) be an instance of U-AnchRobPSP, and let J^* denote the set of jobs that have a predecessor other than s in the precedence graph G . The arc-weights of G^Δ are as follows: the arcs from s to a job $i \notin J^*$ have weight 0, all incoming arcs of t have weight 0, all other arcs have weight 1. Consequently, given $H \subseteq J$, the length of an $s-t$ path in $G^\Delta[H]$ is equal to the number of vertices of J^* in the path. It follows that $L_{G^\Delta[H]}(s, t) \leq M$ if and only if all chains in the subposet $(H \cap J^*, \prec)$ have size at most M . Finally, since G can be any directed acyclic graph, the poset (J^*, \prec) can be any poset. Thus U-AnchRobPSP is equivalent to MaxSubposet.

Theorem 7. *U-AnchRobPSP is solvable in polynomial time.*

Proof. Note first that a subposet in which all chains have size at most M is exactly the union of M antichains (some of the antichains being possibly empty). The problem of finding a max-size union of M antichains in a poset can be solved in polynomial time, through a min cost circulation algorithm (see Theorem 14.8 in (Schrijver (2003))). Using this result and the equivalence between U-AnchRobPSP and MaxSubposet, the result follows. \square

4.3 Dynamic Programming for Budgeted Uncertainty and Series-Parallel Precedence Graphs

This section is devoted to solving AnchRobPSP under budgeted uncertainty for series-parallel precedence graphs. It is assumed that vectors p , $\widehat{\delta}$, and the deadline M have integer values. Series-parallel digraphs are defined recursively as follows. A digraph is *series-parallel* with terminals s and t if one of the three assertions is satisfied:

- Its vertex-set is $\{s, t\}$ and its arc-set is $\{(s, t)\}$;
- (Series composition.) It is formed with two series-parallel digraphs G_1 and G_2 , where terminals t_1 and s_2 have been identified;
- (Parallel composition.) It is formed with two series-parallel digraphs G_1 and G_2 , where the two pairs of terminals s_1 and s_2 , and t_1 and t_2 , have been identified.

In the sequel it is assumed that the precedence graph G is series-parallel with terminals s and t . Given two jobs i and j such that $i \prec j$, let J_{ij} denote the subset of jobs that are successors of i and predecessors of j , with $i, j \notin J_{ij}$, i.e., $J_{ij} = \{k \in J \mid i \prec k \prec j\}$. Let G_{ij} denote the subgraph of G induced by $J_{ij} \cup \{i, j\}$: it is series-parallel with terminals i and j .

The proposed algorithmic scheme relies on the layered graph from Section 2.2. With Theorem 2, a subset $H \subseteq J$ is anchored when there exists a schedule x of $G^{\text{lay}}(H)$ such that $x_t^\Gamma \leq M$. Let us now introduce a value function suitable for a dynamic programming approach. Given a pair of jobs i, j such that $i \prec j$, and $b = (b^\gamma)_{\gamma \in \{0, \dots, \Gamma\}}$ and $b' = (b'^\gamma)_{\gamma \in \{0, \dots, \Gamma\}}$ two vectors of $(\Gamma + 1)$ integer numbers, the *value function* is

$$\begin{aligned}
 V_{ij}(b, b') = \max \quad & a(H) \\
 \text{s.t.} \quad & H \subseteq J_{ij} \\
 & x \text{ schedule of } G_{ij}^{\text{lay}}(H) \\
 & x_i^\gamma = b^\gamma \quad \forall \gamma \in \{0, \dots, \Gamma\} \\
 & x_j^\gamma = b'^\gamma \quad \forall \gamma \in \{0, \dots, \Gamma\}
 \end{aligned}$$

The last two conditions are called *boundary conditions* in the sequel: they enforce the values of starting times of copies of i and j in schedule x . Let $L = L_{G(p+\widehat{\delta})}(s, t)$. Let \underline{b} and \underline{b}' be the vectors defined by $\underline{b}^\gamma = 0$ for every $\gamma \in \{0, \dots, \Gamma\}$, $\underline{b}'^\Gamma = M$, and $\underline{b}'^\gamma = L$ for every $\gamma \in \{0, \dots, \Gamma - 1\}$. The maximum weight of an anchored set is then exactly the value $V_{st}(\underline{b}, \underline{b}')$. Indeed note that the condition $x_t^\gamma \leq L$ for every $\gamma \in \{0, \dots, \Gamma - 1\}$ can be added w.l.o.g. By dynamic programming, the value function will be computed for vectors b, b' in the set $B = \{b = (b^\gamma)_{\gamma \in \{0, \dots, \Gamma\}} \mid b^\gamma \in \{0, \dots, L\} \forall \gamma \in \{0, \dots, \Gamma\}\}$. We now show how to compute the value function in the base case, and prove decomposition properties with respect to series and parallel composition.

Consider first the base case where the graph is just an arc (i, j) .

Lemma 5 (Base case.). *Let G_{ij} be the digraph with vertex-set $\{i, j\}$ and arc-set $\{(i, j)\}$. Let $b, b' \in B$. Then $V_{ij}(b, b') = 0$ if $b'^\gamma - b^\gamma \geq p_i$ for every $\gamma \in \{0, \dots, \Gamma\}$ and $b'^\gamma - b^{\gamma+1} \geq p_i + \widehat{\delta}_i$ for every $\gamma \in \{0, \dots, \Gamma - 1\}$. Otherwise $V_{ij}(b, b') = -\infty$.*

Proof. Proof. Note that $J_{ij} = \emptyset$, hence the value function is 0 if there exists a schedule x of $G_{ij}^{\text{lay}}(\emptyset)$ satisfying the boundary conditions, and $-\infty$ otherwise. Also, the boundary conditions fully define the schedule. Hence the value function is 0 if and only if b and b' satisfy the constraints of $G_{ij}^{\text{lay}}(\emptyset)$, i.e., $b'^\gamma - b^\gamma \geq p_i$ for every $\gamma \in \{0, \dots, \Gamma\}$ and $b'^\gamma - b^{\gamma+1} \geq p_i + \widehat{\delta}_i$ for every $\gamma \in \{0, \dots, \Gamma - 1\}$. \square

Assume G_{ij} is obtained by a parallel composition, w.l.o.g. parallel composition of two series-parallel digraphs G_{ij}^1 and G_{ij}^2 with terminals i and j . Let J_{ij}^1 and J_{ij}^2 denote respectively their sets of inner vertices and V_{ij}^1 and V_{ij}^2 denote their value functions.

Lemma 6 (Parallel composition). *For every $b, b' \in B$, the value function satisfies $V_{ij}(b, b') = V_{ij}^1(b, b') + V_{ij}^2(b, b')$.*

Proof. Proof. Let $H \subseteq J_{ij}$ and (H^1, H^2) be the partition of H defined by $H^1 = H \cap J_{ij}^1$, $H^2 = H \cap J_{ij}^2$. Given x a schedule of G_{ij}^{lay} , let x^1 (resp. x^2) denote the restriction of x to copies of jobs in G_{ij}^1 (resp. G_{ij}^2). It holds that x is a schedule of $G_{ij}^{\text{lay}}(H)$ such that $x_i = b$ and $x_j = b'$ if and only if x^1 and x^2 are schedules of $G_{ij}^{\text{lay}}(H^1)$ and $G_{ij}^{\text{lay}}(H^2)$ respectively, and they satisfy $x_i^1 = b$ and $x_j^1 = b'$, and $x_i^2 = b$ and $x_j^2 = b'$. Hence the value function $V_{ij}(b, b')$ decomposes over the subgraphs G_{ij}^1 and G_{ij}^2 , leading to the desired equality. \square

Assume G_{ij} is obtained by series composition of two series-parallel digraphs G_{ik} and G_{kj} for a given $k \in J_{ij}$.

Lemma 7 (Series composition). *For every $b, b' \in B$, the value function satisfies*

$$V_{ij}(b, b') = \max \left\{ \max_{b'' \in B} \{V_{ik}(b, b'') + V_{kj}(b'', b')\}; \quad a_k + \max_{b'' \in B^{\text{anch}}} \{V_{ik}(b, b'') + V_{kj}(b'', b')\} \right\}$$

where $B^{\text{anch}} = \{b \in B \mid b^\gamma \leq b^\Gamma \ \forall \gamma \in \{0, \dots, \Gamma\}\}$.

Proof. Proof. Let $H \subseteq J_{ij}$ and let $H^1 = H \cap J_{ik}$ and $H^2 = H \cap J_{kj}$. Given x a schedule of G_{ij}^{lay} , let x^1 (resp. x^2) denote the restriction of x to copies of jobs in G_{ik} (resp. G_{kj}). It holds that x is a schedule of $G_{ij}^{\text{lay}}(H)$ such $x_i = b$ and $x_j = b'$ if and only if: x^1 (resp. x^2) is a schedule of $G_{ik}^{\text{lay}}(H^1)$ (resp. $G_{kj}^{\text{lay}}(H^2)$), $x_k^1 = x_k^2 = b''$ for some $b'' \in B$, and either (i) or (ii) is satisfied: (i) $k \notin H$ (ii) $k \in H$, $b'' \in B^{\text{anch}}$. Indeed when $k \in H$, both schedules x^1 and x^2 must satisfy the vertical arcs constraints in k , which is equivalent to $b'' \in B^{\text{anch}}$. Case (i) leads to the first term and case (ii) leads to the second term of the maximum; we thus obtain the desired equality. \square

We can now prove Theorem 8.

Theorem 8. *For fixed uncertainty budget Γ , AnchRobPSP can be solved in $O(mL^{3\Gamma+3})$, where m is the number of arcs in the precedence graph and $L = L_{G(p+\hat{\delta})}(s, t)$.*

Proof. Given a series-parallel digraph G , its *binary decomposition tree* is a binary tree whose leaves are attached to arcs of the digraph, and internal nodes represent series or parallel compositions. The binary decomposition tree of G is computable in linear time (Valdes et al. (1982)) and it has $2m - 1$ nodes, each of them corresponding to a subgraph G_{ij} of G , which is series-parallel with terminals i and j .

Let us now describe the overall algorithm, using the binary decomposition tree of the precedence graph. Nodes of the decomposition tree are considered in a bottom-up fashion. For every subgraph G_{ij} associated with the current node of the tree, compute and store the value $V_{ij}(b, b')$ for every pair of vectors $b, b' \in B$. If G_{ij} is obtained by parallel composition, this can be done in $O(1)$ time (Lemma 6). If G_{ij} is obtained by series composition, this can be done in $O(|B|)$ time (Lemma 7). If G_{ij} is an arc, this can be done in $O(\Gamma)$ time (Lemma 5). Finally return $V_{st}(b, b')$.

The complete table of the value function has $(2m - 1)|B|^2$ entries, and it can be filled in $O(m|B|^2 \max\{|B|, \Gamma\})$. For fixed Γ , since $|B| = O(L^{\Gamma+1})$, the total running time is $O(mL^{3\Gamma+3})$. \square

Note that the values of anchoring weights have no impact on the running time of the proposed algorithm. Note also that the dynamic programming scheme can be refined, e.g., by decreasing the size of the set B ; however this would not change the final pseudo-polynomiality result.

5 Anchor-Robust Solutions for Budgeted Uncertainty

In this section, we highlight the relevance of AnchRobPSP, under box or budgeted uncertainty, based on numerical experiments. We study the impact of parameters (budget Γ , deadline M) on solutions of the problem, and show that AnchRobPSP provides a convenient way to control the price of robustness of a schedule.

5.1 General Settings of Numerical Experiments

For evaluation purpose, different categories of instances are considered, either randomly generated or taken from the literature. Let us first describe the precedence graphs of the instances.

- In **PSP instances**, the precedence graphs are taken from the PSPLib (Kolisch and Sprecher (1996)), a benchmark for the Resource-Constrained Project Scheduling Prob-

lem. We consider 50 instances from the families $j120i$, with $i \in \{1, \dots, 5\}$, which are the largest instances available ($n = 120$ jobs).

- In **ER instances**, the precedence graphs are randomly generated as follows. The number of jobs n ranges in $\{30, 60, 100, 200\}$. Precedence graphs are generated randomly with Erdős-Rényi (**ER**) model, i.e., between each pair of jobs $i < j$ the arc (i, j) is added with probability $pr = \frac{10}{n}$. Processing times of jobs are generated uniformly in $\{1, \dots, 20\}$.
- In **ERC instances**, the precedence graphs are the same as **ER** instances, but with modified processing times so that every job is on a **Critical** path. Processing times of jobs are increased by repeating the two following steps until every job is on a critical path: (i) find a job i with positive margin $m = L_{G(p)}(s, t) - (L_{G(p)}(s, i) + L_{G(p)}(i, t)) > 0$ (ii) increment p_i of a random value in $\{0, \dots, m\}$.

For all instances, the uncertainty set Δ is a budgeted uncertainty set for a given budget Γ , and values $\hat{\delta}_i$ drawn uniformly from $[0, 0.5p_i]$. We considered 50 PSP instances, and we generated 10 ER instances and 10 ERC instances for each value of $n \in \{30, 60, 100, 200\}$, thus resulting in a total of 40 ER instances and 40 ERC instances. Tests were performed with unit anchoring weights. The deadline M is chosen between the minimum makespan $M_{\min} = L_{G(p)}(s, t)$ and the smallest deadline M_{\max} for which all jobs can be anchored. When the deadline M has to be fixed, it is defined arbitrarily as a convex combination of M_{\min} and M_{\max} . Note that the value of Γ or M used in experiments will be specified in the sequel when necessary.

Since PSP instances correspond to a reference for another problem, they have limited relevance for our purpose: namely, they are medium size and they feature precedence graphs with small degree. In contrast, ER instances provide a larger number of jobs, and the expected value of the average degree in the precedence graph is driven by the constant $n \times pr$, which we arbitrarily fixed at 10. Finally ERC instances are motivated by applications to industrial projects, where it can be observed that a lot of paths are critical or almost critical in the baseline instance.

AnchRobPSP was solved either with the MIP formulation from Proposition 1 for $\Gamma < n$; or with Algorithm **A**⁺ for $\Gamma = n$. Algorithms for AnchRobPSP were implemented with Julia 0.6.2, JuMP v0.18.1, and Cplex 12.8 for MIP solving. Numerical experiments were completed on a PC with Intel Core i7-7500U CPU @ 2.70GHz 2.90GHz and 8 Go RAM.

For all numerical results presented in this section, AnchRobPSP or its variant was solved optimally in less than 3 seconds for PSP instances, ER instances, and ERC instances with $n \in \{30, 60\}$. It was solved optimally in less than 15 seconds for ERC instances with $n = 100$,

and less than 4 minutes for ERC instances with $n = 200$. Note that the aim was not to reduce the computation time, but to evaluate the characteristics of the obtained solutions.

5.2 Impact of the uncertainty budget Γ

In this section we consider the case where the decision maker is given the deadline M , and has to decide the value of the uncertainty budget Γ . It is clear that the higher the budget Γ , the less jobs can be anchored, i.e., the optimum of AnchRobPSP is non-increasing with respect to Γ . We now provide experimental results to quantify the impact of Γ on the optimum.

Let us first compare the maximum number of anchored jobs $optAnch(\Gamma)$ for different values of Γ . The value of M is chosen at $\frac{3}{4}M_{\min} + \frac{1}{4}M_{\max}$. Other convex combinations have been tested and lead to similar results. Note that if M is very close to M_{\max} , almost all jobs can be anchored and the impact of the budget on $optAnch$ is very limited. We test small values of Γ , namely $\Gamma \in \{1, 2, 3\}$, and values that are proportional to the number of jobs, namely, $\Gamma \in \{\lceil 5\%n \rceil, 10\%n, 20\%n, 100\%n\}$. Note that this latter case $\Gamma = 100\%n = n$ corresponds to box uncertainty. Results are reported in Tables 1, 2, 3.

		Budget Γ						
		1	2	3	$\lceil 5\%n \rceil$	$10\%n$	$20\%n$	$100\%n$
PSP	j1201	106.5	102.5	100.2	98.6	98.6	98.6	98.6
	j1202	101.0	99.0	99.0	99.0	99.0	99.0	99.0
	j1203	103.8	100.1	98.5	98.1	98.1	98.1	98.1
	j1204	97.0	94.0	94.0	94.0	94.0	94.0	94.0
	j1205	107.0	101.0	99.0	99.0	99.0	99.0	99.0

Table 1: Value $optAnch(\Gamma)$ for different values of budget Γ on PSP instances ($n = 120$)

		Budget Γ						
		1	2	3	$\lceil 5\%n \rceil$	$10\%n$	$20\%n$	$100\%n$
ER	$n = 30$	19.8	17.8	17.0	17.8	17.0	16.6	16.6
	$n = 60$	47.8	44.9	42.9	42.9	41.1	40.5	40.5
	$n = 100$	86.1	82.6	80.1	75.5	73.1	73.0	73.0
	$n = 200$	182.1	177.7	172.7	162.3	161.5	161.5	161.5

Table 2: Value $optAnch(\Gamma)$ for different values of budget Γ on ER instances

n	Budget Γ							
	1	2	3	$\lceil 5\%n \rceil$	$10\%n$	$20\%n$	$100\%n$	
ERC	30	11.6	11.0	10.8	11.0	10.8	10.8	10.8
	60	23.3	22.1	22.1	22.1	22.1	22.1	22.1
	100	42.6	37.0	36.0	35.6	35.6	35.6	35.6
	200	83.8	71.8	70.1	69.8	69.8	69.8	69.8

Table 3: Value $optAnch(\Gamma)$ for different values of budget Γ on ERC instances

Numerical experiments show that the range between $optAnch(1)$ and $optAnch(n)$ remains small. Also, the maximum number of anchored jobs for $\Gamma = n$ is already attained for small values of Γ : on all considered instances, $optAnch(20\%n)$ is equal to $optAnch(n)$. An interpretation is that uncertainty sets with small budget contain already a large enough variety of disruptions for AnchRobPSP.

A related question is whether an optimal solution computed for some budget Γ (e.g., $\Gamma = 1$) will resist to more than Γ disruptions. Given an optimal solution $(x^{\text{opt}}, H^{\text{opt}})$ computed for budget $\Gamma = 1$, we simulate second-stage instances where Γ^{Simu} disruptions occur, with $\Gamma^{\text{Simu}} > \Gamma$, then check whether the schedule $x_H^{\text{opt}} = (x_i^{\text{opt}})_{i \in H^{\text{opt}}}$ can be maintained. We run 1000 simulations and return the percentage of simulations where the answer is yes. Recall that it is easy to check whether the schedule x_H^{opt} can be maintained, with the condition from Lemma 1. Numerical results can be found in Table 4.

	n	Number of disruptions Γ^{Simu}			
		2	3	5	10
PSP	120	99.8%	99.5%	98.5%	93.9%
	30	96.8%	91.1%	74.8%	38.6%
	60	99.0%	96.8%	91.4%	70.0%
	100	99.3%	98.4%	95.1%	81.8%
ER	200	99.9%	99.6%	98.6%	94.0%
	30	98.9%	96.8%	91.1%	76.0%
	60	99.9%	99.5%	98.2%	92.9%
	100	99.8%	99.1%	97.9%	90.2%
ERC	200	99.8%	99.5%	98.4%	93.2%

Table 4: Percentage of simulations where x_H^{opt} can be maintained after Γ^{Simu} disruptions, over 1000 simulations.

We note that the percentage remains over 90% for all instance sets with up to $\Gamma^{\text{Simu}} = 3$ disruptions. Hence the starting times of anchored jobs x_H^{opt} produced by AnchRobPSP for

$\Gamma = 1$ is likely to be maintained after more than Γ disruptions.

Budgeted uncertainty sets with $\Gamma = 1$ may seem too optimistic, because realizations with only one disruption can be considered as relatively favourable. However the results from Table 4 show that the obtained solution resists well in practice to more than one disruption. Moreover Tables 1, 2, 3 show that increasing the budget only decrease moderately the value of $optAnch$. Also $optAnch(n)$ can be easily evaluated with \mathbf{A}^+ . Hence $\Gamma = 1$ can be regarded as a good candidate for choosing the uncertainty budget for AnchRobPSP.

5.3 Trade-off between Makespan and Anchor-Robustness

In this section, the uncertainty set is considered as fixed. We set $\Gamma = 1$ for experiments. The problem AnchRobPSP can be naturally seen as a biobjective optimization problem, where the decision maker must arbitrate between two contradictory objectives, that are the deadline M and the anchoring criterion $|H|$. In this section, we give insights on how AnchRobPSP can be used to achieve the trade-off between makespan and number of anchored jobs.

5.3.1 The price of (anchor-)robustness.

The price of robustness has been introduced in the work of Bertsimas and Sim (2004), as a concept for measuring how solutions get worse (w.r.t. the objective function) when they are made more robust. Let us consider the two robust models for project scheduling from the literature, and compare them with AnchRobPSP in terms of their price of robustness. The robust-static problem looks for the minimum makespan M_{RS} of a schedule feasible for every $\delta \in \Delta$. The robust 2-stage problem from Minoux (2007b) looks for the minimum makespan M_{2S} that can be achieved for every $\delta \in \Delta$ by some schedule x^δ (recall that we fixed $\Gamma = 1$). The associated prices of robustness are $\frac{M_{RS}}{M_{\min}}$ and $\frac{M_{2S}}{M_{\min}}$ respectively, and it holds that $\frac{M_{RS}}{M_{\min}} \geq \frac{M_{2S}}{M_{\min}}$.

Similarly, a *price of anchor-robustness* can be defined as follows. Let the *anchoring target* be some $\alpha \in [0, 100]$: it will correspond to the percentage of anchored jobs required in a solution. Given a schedule x and a set H anchored w.r.t. x , let the *minimum guaranteed second-stage makespan* be the minimum makespan that can be achieved for every $\delta \in \Delta$ by a second-stage schedule x^δ such that $x_H^\delta = x_H$. Let M_α be the minimum guaranteed second-stage makespan over the pairs (x, H) where x schedule of $G(p)$, the set H is anchored w.r.t. x , and $|H| \geq \alpha\%n$. Let then the *price of anchor-robustness* $PoAR_{\alpha\%}$ be the ratio $\frac{M_\alpha}{M_{\min}}$. Note that M_α (and thus $PoAR_{\alpha\%}$) can readily be computed by slightly modifying the MIP from Section 2.2. The definition of $PoAR_{\alpha\%}$ generalizes the price of robustness. Indeed if $\alpha = 0$ (resp. $\alpha = 100$) the minimum guaranteed second-stage makespan $M_{\alpha=0}$ (resp. $M_{\alpha=100}$) is exactly the robust makespan M_{2S} (resp. M_{RS}) computed by the robust 2-stage (resp. robust-static) problem. Hence $PoAR_{0\%} = \frac{M_{2S}}{M_{\min}}$ and $PoAR_{100\%} = \frac{M_{RS}}{M_{\min}}$.

Consequently, by varying the anchoring target α in $[0, 100]$, it is possible to tune the price of (anchor-)robustness, and get intermediary options between robust 2-stage solutions and robust-static solutions, where all jobs are anchored. Numerical results are provided in Table 5. They illustrate the classical fact that robust 2-stage and robust-static problems yields very different outputs in terms of price of robustness: e.g., for ER instances with $n = 30$ jobs, opting for a robust-static problem causes an increase of +28% of the makespan w.r.t. M_{\min} , vs. only +4.7% for the robust 2-stage problem. In between are solutions of AnchRobPSP for different values of the anchoring target. Note that it is possible to obtain solutions with anchored jobs without increasing the price of robustness, e.g., for PSP instances, it is possible to get 50% of jobs anchored while keeping the robust 2-stage makespan.

		Anchoring target α						
		0	10	20	50	80	90	100
	n	(2-stage)						(static)
PSP	120	1.049	1.049	1.049	1.049	1.060	1.119	1.313
	30	1.047	1.047	1.047	1.051	1.149	1.202	1.280
	60	1.044	1.044	1.044	1.044	1.089	1.164	1.280
	100	1.037	1.037	1.037	1.037	1.052	1.129	1.293
	200	1.037	1.037	1.037	1.037	1.038	1.076	1.274
ERC	30	1.149	1.149	1.154	1.192	1.258	1.295	1.366
	60	1.167	1.167	1.171	1.197	1.266	1.304	1.379
	100	1.169	1.169	1.171	1.189	1.257	1.296	1.388
	200	1.208	1.208	1.208	1.222	1.276	1.313	1.410

Table 5: Values of the price of anchor-robustness $PoAR_{\alpha\%}$ for various values of the anchoring target α .

5.3.2 A biobjective perspective on makespan and anchor-robustness.

When solving AnchRobPSP, it might be in practice that the deadline M is dictated by exogenous factors and given to the decision maker. If so, the value of M is fixed regardless of its impact on the anchoring criterion. Our claim is that information on the impact of M on the anchoring criterion may be of great interest for the decision maker, who could consider asking for a revision of the value of M .

By solving AnchRobSP for various values of M we obtain a Pareto front, where Pareto-optimal solutions of AnchRobPSP are represented in the solution space for the two criteria. It is clear that the optimal value of AnchRobPSP is non-decreasing w.r.t. M : for a very short deadline M , only a small subset of jobs can be anchored; if M is sufficiently large all jobs can be anchored. An example is presented in Figure 3 on an ERC instance with $n = 60$

jobs. While robust-static problem only provides one point of the Pareto front, AnchRobPSP provides a wider variety of solutions, among which the decision maker may pick its preferred one.

Examples of Gantt charts of four Pareto-optimal solutions are provided in Figure 4. In every Gantt chart, jobs of H are in black. Solution S_1 has makespan M_{\min} , hence only a few jobs can be anchored (and since it is an ERC instance, only jobs with no predecessors can be anchored). Solution S_4 is a robust-static schedule. Solutions S_2 and S_3 are other possible options. Note that the position of S_3 in the Pareto front indicates that any solution with more anchored jobs than S_3 has a makespan at least 9 units greater than the makespan of S_3 .

This approach gives a new way to fix the deadline M , by choosing an anchoring target α , and setting M to the smallest value for which there exists a Pareto-optimal solution with at least $\alpha\%$ anchored jobs.

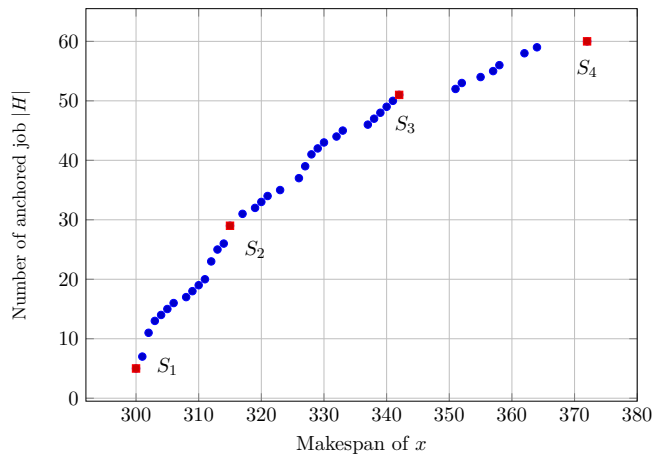


Figure 3: Pareto front makespan/anchoring criterion for an ERC instance with $n = 60$.

6 Conclusion

This study of AnchRobPSP was carried out for a variety of uncertainty sets, with an emphasis on budgeted uncertainty. Recall that in robust optimization, the robust 2-stage problem under budgeted uncertainty is commonly used as it allows for a reduced conservatism w.r.t. the static variant. In this work we showed that AnchRobPSP is NP-hard for budgeted uncertainty. This is in contrast with the complexity of the robust 2-stage problem, which is polynomial. The increase in complexity stems from binary decisions on whether a job is anchored or not, which have to be made in AnchRobPSP because of the deadline and the uncertainty set. Even for box uncertainty, AnchRobPSP differs from robust problems. Indeed the latter problems can usually be summed up to solving the problem for the worst-case value of parameters, while

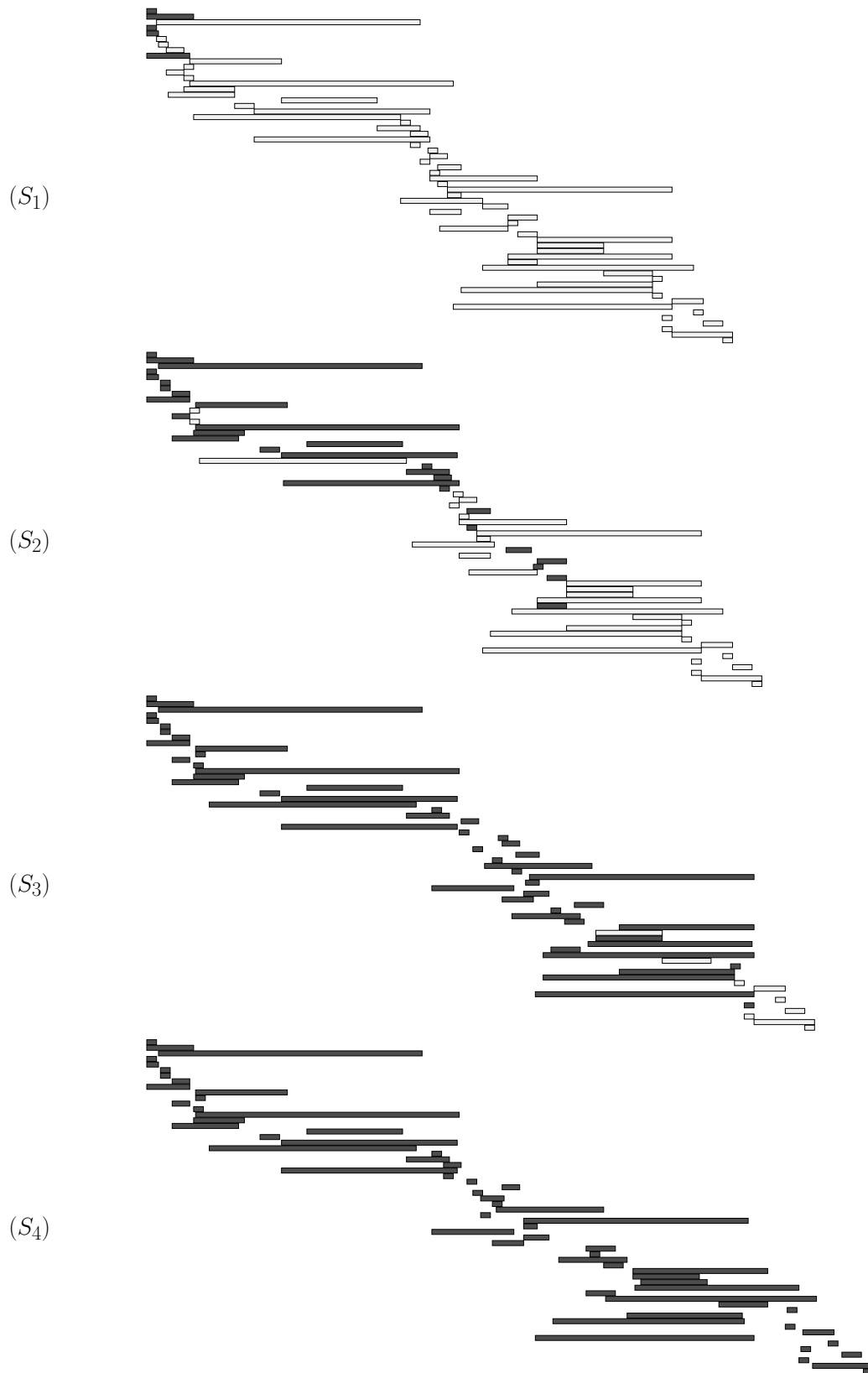


Figure 4: Gantt charts of four solutions from the Pareto front. Anchored jobs are in black.

AnchRobPSP requires a specific algorithm (Algorithm \mathbf{A}^+). Finally, the interest of small uncertainty budget for AnchRobPSP, e.g., $\Gamma = 1$, was illustrated in numerical experiments. Indeed even for small budget, an optimal solution of AnchRobPSP is to anchor all jobs whenever the deadline M allows. AnchRobPSP can also be used to define the minimum makespan to anchor a subset of jobs under a $\Gamma = 1$ budgeted uncertainty set.

In the line of the complexity analysis of AnchRobPSP, a perspective is to study the open question of AnchRobPSP complexity when the values L_{ij}^Δ are part of the input (or computable in polynomial time) and with unit anchoring weights. From a practical viewpoint, a perspective is to tackle large-scale instances: indeed industrial instances may feature thousands of maintenance operations. Algorithms for AnchRobPSP could also be integrated in a decision making tool, to assist planning managers in making the trade-off between multiple criteria, including makespan and anchoring criterion. Anchor-robust variants of other problems could also be formulated and studied. This includes, e.g., project scheduling problems with resource constraints, or other applications beyond scheduling. The definition of an anchoring criterion to evaluate which decisions are guaranteed, seems relevant in practice in a large variety of combinatorial problems.

References

- Pascale Bendotti, Philippe Chrétienne, Pierre Fouilhoux, and Alain Quilliot. Anchored reactive and proactive solutions to the CPM-scheduling problem. *European Journal of Operational Research*, 261:67–74, 2017.
- Dimitris Bertsimas and Melvyn Sim. The price of robustness. *Operations Research*, 52:35–53, 2004.
- Endre Boros, Konrad Borys, Khaled M. Elbassioni, Vladimir Gurvich, and Gábor Rudolf. Inapproximability bounds for shortest-path network interdiction problems. 2005.
- Gianlorenzo D’Angelo, Gabriele Di Stefano, Alfredo Navarra, and Cristina Pinotti. Recoverable robust timetables: An algorithmic approach on trees. *IEEE Transactions on Computers*, 60:433–446, 2011.
- Virginie Gabrel, Cécile Murat, and Aurélie C. Thiele. Recent advances in robust optimization: An overview. *European Journal of Operational Research*, 235:471–483, 2014.
- Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Co., New York, NY, USA, 1979.
- Willy Herroelen and Roel Leus. Project scheduling under uncertainty: Survey and research potentials. *European Journal of Operational Research*, 165:289–306, 2002.
- Willy Herroelen and Roel Leus. The construction of stable project baseline schedules. *European Journal of Operational Research*, 156(3):550–565, 2004.
- Eitan Israeli and R. Kevin Wood. Shortest-path network interdiction. *Networks*, 40(2):97–111, 2002.
- Rainer Kolisch and Arno Sprecher. PSPLIB – a project scheduling problem library. *European Journal of Operational Research*, 96:205–216, 1996.

- Christian Liebchen, Marco Lübbecke, Rolf Möhring, and Sebastian Stiller. The concept of recoverable robustness, linear programming recovery, and railway applications. *Robust and Online Large-Scale Optimization*, 5868:1–27, 2009.
- Michel Minoux. Models and algorithms for robust PERT scheduling with time-dependent task durations. *Vietnam Journal of Mathematics*, 35, 01 2007a.
- Michel Minoux. Duality, Robustness, and 2-stage robust LP decision models. Application to Robust PERT Scheduling. *Annales du LAMSADE N.7*, 2007b.
- Artur Alves Pessoa, Luigi Di Puglia Pugliese, Francesca Guerriero, and Michael Poss. Robust constrained shortest path problems under budgeted uncertainty. *Networks*, 66(2):98–111, 2015.
- Alexander Schrijver. *Combinatorial Optimization - Polyhedra and Efficiency*. Springer, 2003.
- A. L. Soyster. Technical note – convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations Research*, 21(5):1154–1157, 1973.
- J. Valdes, R. Tarjan, and E. Lawler. The recognition of series parallel digraphs. *SIAM Journal on Computing*, 11(2):298–313, 1982.