



HAL
open science

Dominance-based linear formulation for 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. Dominance-based linear formulation for the Anchor-Robust Project Scheduling Problem. 2020. <hal-02938158v1>

HAL Id: hal-02938158

<https://hal.science/hal-02938158v1>

Preprint submitted on 14 Sep 2020 (v1), last revised 22 Jun 2021 (v2)

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.



HAL Authorization

Dominance-based linear formulation for the Anchor-Robust Project Scheduling Problem

Pascale Bendotti^{1,2}, Philippe Chrétienne², Pierre Foulhoux², and Adèle Pass-Lanneau^{1,2}

¹ EDF R&D, F-91120 Palaiseau, France,

{pascale.bendotti, adele.pass-lanneau}@edf.fr

² Sorbonne Université, CNRS, LIP6, F-75005 Paris, France,

{philippe.chretienne, pierre.foulhoux}@lip6.fr

September 14, 2020

Abstract

In project scheduling under processing times uncertainty, the Anchor-Robust Project Scheduling Problem is to find a baseline schedule of bounded makespan and a max-weight subset of jobs whose starting times are guaranteed. The problem was proven NP-hard even for budgeted uncertainty. In the present work we design mixed-integer programming (MIP) formulations that are valid for a variety of uncertainty sets encompassing budgeted uncertainty. A new dominance among solutions is proposed, resulting into an MIP formulation. This dominance-based formulation is shown to be tighter than all previously known formulations. We further study the combinatorial structure of the problem. We show that the dominance-based formulation yields a polyhedral characterization in non-trivial cases of budgeted uncertainty. In numerical experiments we investigate how the formulation performs on instances around the polyhedral characterization cases, for both budgeted uncertainty sets and more elaborate uncertainty sets involving several budgets.

Keywords: project scheduling; combinatorial optimization; mixed-integer programming; robust 2-stage optimization; polyhedral characterization.

1 Introduction

Consider a set of n jobs J to be scheduled under precedence constraints. The *precedence graph* is a directed acyclic graph $G(p) = (\bar{J}, \mathcal{A}, p)$, where $\bar{J} = J \cup \{s, t\}$ and s (resp. t) is a dummy job representing the beginning (resp. end) of the project. Processing times of jobs are $p \in \mathbb{R}_+^J$ and every arc $(i, j) \in \mathcal{A}$ has length p_i , with $p_s = 0$. A schedule of $G(p)$ is a vector $x \in \mathbb{R}_+^{\bar{J}}$ of starting times such that $x_s = 0$ and $x_j - x_i \geq p_i$ for every arc (i, j) of $G(p)$. Finding a schedule with minimum *makespan* $x_t = \min_{i \in J} x_i + p_i$ is a classical polynomial-time solvable problem [16].

In practice processing times may be uncertain parameters, which calls for robust approaches. We consider that jobs have nominal processing times p and the real processing times may be $p + \delta$, where δ lies in an uncertainty set $\Delta \subset \mathbb{R}_+^J$. For a given uncertainty set, different robust approaches can be considered. A first approach from the *robust-static* framework [19] is to find a schedule $x \in \mathbb{R}_+^{\bar{J}}$ such that x is a schedule of $G(p + \delta)$ for every $\delta \in \Delta$, and so that the makespan of x is minimized. The robust-static approach is known to be overly conservative,

i.e., it produces a schedule with a very large makespan. An alternate approach from *robust 2-stage optimization*, also called adjustable in [2], is to find the minimum value B such that for every $\delta \in \Delta$ there exists a schedule $x^\delta \in \mathbb{R}_+^J$ of $G(p+\delta)$ such that $x_t^\delta \leq B$. The robust 2-stage approach often gives a worst-case makespan that is significantly lower than the robust-static makespan. This makes the robust 2-stage approach appealing. It was studied in [15] for project scheduling. A major drawback of the robust 2-stage approach is that the schedule depends on the uncertainty realization. In other words, starting times cannot be decided before δ is known.

Project scheduling has a large variety of practical applications such as planning industrial activities. The durations of activities are often not known exactly or they can change over time: this calls for the study of project scheduling under processing times uncertainty. It is often necessary to compute a schedule in advance, called *baseline schedule* [12], while taking into account processing times uncertainty. The precomputation of a baseline schedule is common practice, especially when scheduling activities requires preparation and coordination with other entities to secure the availability of staff or specific equipment. The makespan of the baseline schedule is a major criterion; in some cases, the project must be scheduled so that it completes before a deadline. Moreover if processing times are disrupted and the schedule must be revised, it is important to stick to the baseline schedule [13]. As some activities may be difficult to reschedule, it is interesting to guarantee their starting times in the baseline schedule against uncertainty realizations.

The Anchor-Robust Project Scheduling Problem. A new robust problem was introduced in [3] to find a baseline schedule that has bounded makespan, along with a guarantee of some starting times. Formally, given a schedule x of $G(p)$ and uncertainty set Δ , a subset of jobs $H \subseteq J$ is *x -anchored* if for every $\delta \in \Delta$ there exists x^δ schedule of $G(p+\delta)$ such that $x_i^\delta = x_i$ for every $i \in H$. Given a *deadline* $M \geq 0$, and *anchoring weights* $w \in \mathbb{R}_+^J$, the Anchor-Robust Project Scheduling Problem (AnchRob) is to find a pair (x, H) maximizing the anchoring weight $\sum_{i \in H} w_i$, with x a schedule of $G(p)$ with makespan at most M , and $H \subseteq J$ an x -anchored subset of jobs. This corresponds to the following program:

$$\begin{aligned}
 \text{(AnchRob):} \quad & \max \quad \sum_{i \in H} w_i \\
 & \text{s.t.} \quad x \text{ schedule of } G(p) \\
 & \quad \quad x_t \leq M \\
 & \quad \quad H \subseteq J: \forall \delta \in \Delta, \exists x^\delta \text{ schedule of } G(p+\delta) \\
 & \quad \quad \quad \text{s.t. } x_i = x_i^\delta \quad \forall i \in H
 \end{aligned}$$

If there is a solution (x, J) of (AnchRob), i.e., a solution where all jobs are anchored, then x is a solution of the robust-static problem; otherwise, it means that no robust-static solution respects deadline M . (AnchRob) is a 2-stage problem: x and H are first-stage decisions; x^δ are second-stage decisions. Schedule x is thus a baseline schedule which may be adjusted in second stage by revising starting times of non-anchored jobs only.

Note that the proposed program is related to max/min/max problems arising usually in robust 2-stage optimization. Given a schedule x of $G(p)$, a subset $H \subseteq J$, and an uncertainty realization $\delta \in \Delta$, the second-stage or recourse problem of (AnchRob) is to maximize $\sum_{i \in H} w_i$ subject to x^δ being a schedule of $G(p+\delta)$ with $x_i = x_i^\delta$ for every $i \in H$. Hence this recourse problem reduces to deciding the existence of such an x^δ .

Let us present several uncertainty sets of interest for (AnchRob). This includes *budgeted uncertainty sets* [7] where at most Γ processing times deviate from the nominal values p . A special case is *box uncertainty sets*, in which every processing time deviates from its nominal value within an interval, corresponding to uncertainty budget $\Gamma = n$. Another special case is *1-disruption uncertainty sets*, in which one deviation of fixed length occurs to any job in the project. We also consider more elaborate uncertainty sets, corresponding to unions or

intersections of budgeted uncertainty sets. Such uncertainty sets account, e.g., for the case of several uncertainty budgets on subsets of jobs.

Robust 2-stage problems are often considered challenging and computationally intensive [9, 11]. A compact MIP formulation for robust 2-stage problems is often not known, and proposed exact approaches rely on decomposition methods, see, e.g., [8, 1, 20]. Another trend is to consider approximations. With *decision rules* introduced in [2] it is assumed that second-stage variables have a fixed dependency (e.g., affine) on the uncertainty realization. It yields tractable approximations of the original robust 2-stage problem. With *k-adaptability* introduced by [6] a fixed number of recourse solutions is determined in first stage and the recourse problem is to choose the best among them. The focus of the present work is to investigate exact approaches for the (AnchRob) problem, and especially compact formulations, which – to the best of our knowledge – have been scarcely studied for robust 2-stage problems.

Robust 2-stage problems including a baseline solution in first stage and a feature to keep decisions unchanged were considered under the name of *recoverable robustness* [14, 10] or *robust optimization with incremental recourse* [18]. When decisions are represented by continuous variables, different measures can be used to account for the stability of decisions. For project scheduling, the expected absolute gap between starting times was used as stability measure in [13]. The anchoring criterion was defined as the number of identical starting times in [5]. This criterion was studied for rescheduling problems in [4]. The (AnchRob) problem was introduced in [3], and it was proven NP-hard even for budgeted uncertainty. For budgeted uncertainty, a compact MIP reformulation was obtained, denoted by (Lay). It is based on a so-called layered graph and a dedicated analysis of the problem in this special case. This MIP formulation is inherent to budgeted uncertainty, and thus not applicable to other uncertainty sets.

In this work, we investigate linear formulations for (AnchRob) that are valid for a variety of uncertainty sets encompassing budgeted uncertainty. The starting point is to precompute the worst-case value over Δ of the longest $i-j$ path for every pair of jobs i, j , and to use these values in a linear formulation. We exhibit a dominance property among schedules and derive a linear formulation from it. This dominance-based linear formulation, called (Dom), improves over a naive linear formulation. In the case of budgeted uncertainty, (Dom) is also stronger than the known formulation (Lay). A polyhedral study is carried out to highlight how formulation (Dom) captures the combinatorial structure of the problem. We prove that (Dom) yields a complete polyhedral characterization in two special cases: box uncertainty, and 1-disruption uncertainty for precedence graphs where all $s-t$ paths are longest $s-t$ paths. Finally numerical experiments give evidence that (Dom) performs well for budgeted uncertainty, even for instances that do not match the polyhedral characterization cases. (Dom) is also capable of solving the problem for uncertainty sets where no MIP formulation was previously investigated, e.g., in the case of several budgets.

In Section 2 notation and preliminaries on problem (AnchRob) are given. In Section 3 the dominance-based linear formulation is presented, and shown to be stronger than other known formulations. In Section 4 polyhedral characterizations are proven for box uncertainty, and 1-disruption uncertainty. In Section 5 numerical results are presented.

2 Preliminaries

2.1 General notation

Let $G(p) = (\bar{J}, \mathcal{A}, p)$ be the precedence graph. It is assumed that there is an arc (s, i) (resp. an arc (i, t)) for every job $i \in J$ without predecessor (resp. without successor) in J . Let \prec

denote the partial order on \bar{J} defined by $i \prec j$ if there exists an i - j path in G . Given $i, j \in \bar{J}$, $i \prec j$, let $L_{G(p)}(i, j)$ be the length of the longest i - j path in $G(p)$. In the sequel we will use the shorthand notation $L_{ij}^0 = L_{G(p)}(i, j)$. The minimum makespan of a schedule of $G(p)$ is then L_{st}^0 . The *earliest schedule* defined by $x_i = L_{si}^0$ for every $i \in J$ is a schedule with minimum makespan. Let $X^{\leq M}$ denote the set of schedules of $G(p)$ with makespan at most M . A longest s - t path is a *critical path*. The precedence graph $G(p)$ is *quasi-critical* if every job $i \in J$ belongs to a critical path, i.e., $L_{si}^0 + L_{it}^0 = L_{st}^0$. The precedence graph $G(p)$ is *critical* if all s - t paths in $G(p)$ are critical, i.e., they have length L_{st}^0 . Note that if $G(p)$ is critical then it is quasi-critical. The uncertainty set Δ is assumed to be a subset of \mathbb{R}_+^J . For uncertainty set Δ , let us define the value

$$L_{ij}^\Delta = \max_{\delta \in \Delta} L_{G(p+\delta)}(i, j)$$

for every $i, j \in \bar{J}$, $i \prec j$. Then L_{ij}^Δ is the *worst-case longest path value* between job i and job j .

2.2 Uncertainty sets

Note first that some assumptions on Δ can be made without loss of generality. The uncertainty set can be assumed to be convex [2]. The uncertainty set can also be assumed to be down-monotone, i.e., if $\delta \in \Delta$ and $\delta' \leq \delta$ then $\delta' \in \Delta$. Indeed it directly follows from the fact that if x^δ is a schedule of $G(p + \delta)$, then it is a schedule of $G(p + \delta')$ for every $\delta' \leq \delta$. In the present work, considered uncertainty sets will be polyhedra, and w.l.o.g. down-monotone.

Let us now present some uncertainty sets of interest.

Set Δ is a *box uncertainty set* if $\Delta = \{(\delta_i)_{i \in J} : 0 \leq \delta_i \leq \hat{\delta}_i \forall i \in J\}$ with $\hat{\delta} \in \mathbb{R}_+^J$, i.e., it is a cartesian product of intervals. Then $\hat{\delta}$ is a greatest element of Δ in the sense that $\delta \leq \hat{\delta}$ for every $\delta \in \Delta$. Note that if Δ is any set with greatest element $\hat{\delta}$, then w.l.o.g. it can be assumed to be down-monotone, and thus equal to the box with greatest element $\hat{\delta}$.

Set Δ is a *budgeted uncertainty set* if $\Delta = \{(\hat{\delta}_i u_i)_{i \in J} : u \in [0, 1]^J, \sum_{i \in J} u_i \leq \Gamma\}$, with *deviation* $\hat{\delta} \in \mathbb{R}_+^J$ and *uncertainty budget* $\Gamma \in \{1, \dots, |J|\}$. The extreme points of such Δ correspond to budgeted uncertainty as defined in the seminal work of [7], where Γ is the number of processing times that may deviate from their nominal value. Box uncertainty is the special case of budgeted uncertainty where $\Gamma = |J|$.

Set Δ is a *1-disruption uncertainty set* if it is a budgeted uncertainty set with unit budget $\Gamma = 1$ and uniform deviation, i.e., $\hat{\delta}_i = \hat{\delta}_0$ for every $i \in J$. Extreme points of a 1-disruption uncertainty set represent the situation where one event of fixed – possibly large – deviation $\hat{\delta}_0$ may happen anywhere in the project. Then the processing time of one job is increased by fixed amount $\hat{\delta}_0$.

Let us now present more elaborate uncertainty sets, built as unions or intersections of budgeted uncertainty sets.

Set Δ is a *partition-budgeted uncertainty set* if $\Delta = \{(\hat{\delta}_i u_i)_{i \in J} : u \in [0, 1]^J, \sum_{i \in J^k} u_i \leq \Gamma^k \forall k \in \{1, \dots, m\}\}$ where (J^1, \dots, J^m) is a partition of J and $\Gamma^k \in \{1, \dots, |J^k|\}$ for every $k \in \{1, \dots, m\}$. Each group J^k is associated with its own uncertainty budget Γ^k . It holds that $\Delta = \bigcap_{1 \leq k \leq m} \Delta^k$ where $\Delta^k = \{(\hat{\delta}_i u_i)_{i \in J} : u \in [0, 1]^J, \sum_{i \in J^k} u_i \leq \Gamma^k\}$ for every $k \in \{1, \dots, m\}$. Distinct uncertainty budgets on disjoint subsets of the partition are relevant when deviations of jobs from different subsets are uncorrelated. A special case of interest is to consider a partition (J^1, J^2) where jobs of the first subset J^1 are associated with small deviations but large uncertainty budget Γ^1 ; jobs of the second subset J^2 are associated with large deviations but small uncertainty budget Γ^2 .

Set Δ is a *mixed-budgeted uncertainty set* if $\Delta = \bigcup_{1 \leq k \leq m} \Delta^k$ where Δ^k are budgeted uncertainty sets. Consider the following special case, where Δ^1 is defined by deviation $\widehat{\delta}$ and budget Γ^1 , and Δ^2 is defined by deviation $\tau\widehat{\delta}$ for a given $\tau \in [0, 1]$, and budget $\Gamma^2 > \Gamma^1$. Then $\Delta = \Delta^1 \cup \Delta^2$ supports two kinds of uncertainty realizations corresponding either to a large number of small deviations (i.e., $\delta \in \Delta^2$) or a small number of large deviations (i.e., $\delta \in \Delta^1$). For uniform deviation, it holds that $\Delta^1 \subsetneq \Delta$ whenever $\tau\Gamma^2 > \Gamma^1$.

All these uncertainty sets are defined implicitly through inequalities or set operations. Note that the uncertainty set may also be given explicitly as the convex hull of a discrete set of points. Given a set S of scenarii, for every $s \in S$ the uncertainty realization is some $\delta^s \in \mathbb{R}_+^J$, and $\Delta = \text{conv}\{\delta^s, s \in S\}$. Any implicitly-defined set Δ can be written under that form by enumerating its extreme points, but then the input size is exponentially increased.

2.3 Assumption on longest paths computation

Computing worst-case longest path values L_{ij}^Δ can be an NP-hard problem for polyhedral set Δ defined by inequalities [3].

However, efficient algorithms can be designed for some uncertainty sets. For box uncertainty, the computation is straightforward since $L_{ij}^\Delta = L_{G(p+\widehat{\delta})}(i, j)$. For budgeted uncertainty sets, and thus 1-disruption uncertainty sets, the worst-case longest path values L_{ij}^Δ can be computed in polynomial time by dynamic programming [15]. The algorithm is linear in $\Gamma|\mathcal{A}|$. For partition-budgeted uncertainty sets, the L_{ij}^Δ values can be computed using a straightforward generalization of the dynamic programming of [15] which is linear in $(\prod_{1 \leq k \leq m} \Gamma^k)|\mathcal{A}|$. For mixed-budgeted uncertainty sets, the L_{ij}^Δ values can be easily obtained since $L_{ij}^\Delta = \max_{1 \leq k \leq m} L_{ij}^{\Delta^k}$ for every $i, j \in \bar{J}$. For explicitly-defined uncertainty sets, values L_{ij}^Δ can be computed in polynomial time in the number of scenarii $|S|$, since $L_{ij}^\Delta = \max_{s \in S} L_{G(p+\delta^s)}(i, j)$. This relies on the property that $L_{ij}^\Delta = L_{ij}^{\text{conv}(\Delta)}$ for any Δ , as proven in [3].

In the sequel, the main assumption is that the L_{ij}^Δ values have been precomputed. Thus they will appear as coefficients of constraints in the proposed mixed-integer programming formulations.

2.4 Anchored sets

Let us finally give some preliminaries related to linear formulations for (AnchRob). Recall that a solution of (AnchRob) is a pair (x, H) with a schedule x and a subset H of jobs that is x -anchored. Fix x a schedule of $G(p)$. It was shown in [3] that a set H is x -anchored if and only if $x_j - x_i \geq L_{ij}^\Delta$ for every $i \in H \cup \{s\}$, $j \in H$, $i \prec j$. Let \widetilde{G}_H denote the precedence graph obtained from $G(p)$ by introducing additional arcs (i, j) , $i \in H \cup \{s\}$, $j \in H$, $i \prec j$ with arc-lengths L_{ij}^Δ . Then

Proposition 1 ([3]). *Let $H \subseteq J$. Vector x is a schedule of $G(p)$ such that H is x -anchored if and only if x is a schedule of \widetilde{G}_H .*

Let us now define anchored sets, independently from a given baseline schedule. A subset $H \subseteq J$ is *anchored* if there exists a schedule x of $G(p)$ with makespan at most M such that H is x -anchored. An issue is then to retrieve a baseline schedule x for which H is x -anchored. Consider z the earliest schedule of \widetilde{G}_H : then by Proposition 1, if H is an anchored set then H is z -anchored. Given $H \subseteq J$, let χ^H denote the incidence vector of H . Let $\mathcal{H} = \{\chi^H : H \text{ anchored set}\}$,

and $\mathcal{Q} = \text{conv}(\mathcal{H})$. Note that (AnchRob) reduces to finding a max-weight anchored set, i.e., maximizing $\sum_{i \in J} w_i h_i$ for $h \in \mathcal{H}$, or equivalently, for $h \in \mathcal{Q}$.

Let us give some definitions related to mixed-integer programming formulations for (AnchRob). Considered formulations for (AnchRob) involve binary anchoring variables $h \in \{0, 1\}^J$ to indicate if jobs are in the anchored set, and continuous variables, say $x \in \mathbb{R}^q$. A formulation for (AnchRob) is defined by a polyhedron $\mathcal{P} \subseteq \mathbb{R}^q \times [0, 1]^J$ and integrality constraints $h \in \{0, 1\}^J$, so that the feasible set of the formulation is $\mathcal{P} \cap (\mathbb{R}^q \times \{0, 1\}^J)$. Given $\mathcal{F} \subseteq \mathbb{R}^q \times \mathbb{R}^J$, let $\text{Proj}_h(\mathcal{F}) = \{h \in \mathbb{R}^J : \exists(x, h) \in \mathcal{F}\}$ denote its projection on h variables. A formulation is *valid for (AnchRob)* if $\text{Proj}_h(\mathcal{P}) \cap \{0, 1\}^J = \mathcal{H}$. A formulation *yields a polyhedral characterization for (AnchRob)* if $\text{Proj}_h(\mathcal{P}) = \mathcal{Q}$. Given two polyhedra \mathcal{P}_1 and \mathcal{P}_2 , formulation associated with \mathcal{P}_1 is *stronger* than formulation associated with \mathcal{P}_2 if $\text{Proj}_h(\mathcal{P}_1) \subseteq \text{Proj}_h(\mathcal{P}_2)$.

3 Linear formulations for (AnchRob)

In this section, we establish linear formulations for (AnchRob) using L_{ij}^Δ values as coefficients. In Section 3.1 a naive formulation is given as a benchmark. In Section 3.2 the main dominance property is proven. In Section 3.3 a formulation is derived from the dominance property. In Section 3.4 it is proven stronger than other known formulations.

3.1 A naive formulation

Consider schedule continuous variables x_j , $j \in \bar{J}$, and anchoring binary variables $h_j \in \{0, 1\}$, $j \in J$. Vector h is the incidence vector of the anchored set H . A formulation for (AnchRob) requires constraints to enforce that H is an x -anchored set. The characterization of Proposition 1 suggests a quadratic constraint

$$x_j - x_i \geq L_{ij}^\Delta h_i h_j \quad \forall i, j \in \bar{J}, i \prec j \quad (1)$$

to represent precedence constraints of the graph \tilde{G}_H . Note that for dummy jobs s and t there is no h_j decision variable but we set $h_s = 1$ and $h_t = 0$ for the ease of notation. For validity, it is sufficient to check that for each $i \prec j$, if $h_i = 0$ or $h_j = 0$ then inequality (1) is valid. Indeed it reduces to $x_j - x_i \geq 0$ which holds for every $i \prec j$ for every schedule x of $G(p)$.

Applying a standard linearization technique, a linear formulation (Std) can be obtained by replacing constraint (1) by the following linear inequality (2)

$$\begin{aligned} \text{(Std): } \quad & \max \quad \sum_{i \in J} w_i h_i \\ & \text{s.t.} \quad x_j - x_i \geq p_i \quad \forall (i, j) \in \mathcal{A} \\ & \quad \quad x_t \leq M \\ & \quad \quad x_j - x_i \geq L_{ij}^\Delta (h_i + h_j - 1) \quad \forall i, j \in \bar{J}, i \prec j \\ & \quad \quad x_j \geq 0 \quad \forall j \in \bar{J} \\ & \quad \quad h_j \in \{0, 1\} \quad \forall j \in J \end{aligned} \quad (2)$$

It is easy to check that when $h_i = 0$ or $h_j = 0$, (2) induces a valid constraint.

3.2 A dominance property

Let $H \subseteq J$ be a subset of jobs. Let us define the set of all baseline schedules x such that (x, H) is feasible for (AnchRob), i.e., $X^{\leq M}(H) = \{x \in X^{\leq M} : H \text{ } x\text{-anchored}\}$. Note that (AnchRob) problem is to maximize the weight of a set H such that $X^{\leq M}(H) \neq \emptyset$. By Proposition 1,

$$X^{\leq M}(H) = \{x \in X^{\leq M} : x_j - x_i \geq L_{ij}^\Delta \quad \forall i \in H \cup \{s\}, j \in H, i \prec j\}.$$

Let us now define a set of baseline schedules where the same inequality is imposed, but on pairs i, j with $i \in J \cup \{s\}$:

$$Z^{\leq M}(H) = \{z \in X^{\leq M} : z_j - z_i \geq L_{ij}^\Delta \quad \forall i \in J \cup \{s\}, j \in H, i \prec j\}.$$

Theorem 1. *Set $Z^{\leq M}(H)$ is dominant, in the sense that $Z^{\leq M}(H) \subseteq X^{\leq M}(H)$, and $X^{\leq M}(H) \neq \emptyset$ implies $Z^{\leq M}(H) \neq \emptyset$.*

Proof. Since $H \subseteq J$, the definition of set $Z^{\leq M}(H)$ contains more constraints than that of set $X^{\leq M}(H)$. Hence $Z^{\leq M}(H) \subseteq X^{\leq M}(H)$. Let us prove $X^{\leq M}(H) \neq \emptyset \implies Z^{\leq M}(H) \neq \emptyset$. Note that $X^{\leq M}(H)$ is exactly the set of schedules of \tilde{G}_H with makespan at most M . Let z be the earliest schedule of \tilde{G}_H . By assumption, there exists a schedule in $X^{\leq M}(H)$, thus $z_t \leq M$. Let $i \in J \cup \{s\}$ and $j \in H$. Let us show that $z_j - z_i \geq L_{ij}^\Delta$ holds, even for $i \notin H \cup \{s\}$. Consider a longest $s-i$ path $P_{s_i}^*$ in \tilde{G}_H , and let $k \in H \cup \{s\}$ be the last vertex of $H \cup \{s\}$ on this path. Then $z_i = L_{\tilde{G}}(s, i) = L_{\tilde{G}}(s, k) + L_{\tilde{G}}(k, i) = z_k + L_{\tilde{G}}(k, i)$. The subpath of $P_{s_i}^*$ from k to i is a longest $k-i$ path in \tilde{G}_H . Since it has no vertex in $H \cup \{s\}$ except k , it uses no additional arc and its length is $L_{\tilde{G}}(k, i) = L_{ki}^0$. It comes $z_k - z_i = -L_{ki}^0$. Since $z \in X^{\leq M}(H)$, $z_j - z_k \geq L_{kj}^\Delta$. Then $z_j - z_i = (z_j - z_k) + (z_k - z_i) \geq L_{kj}^\Delta - L_{ki}^0$. Also, for $k \prec i \prec j$, it holds that $L_{kj}^\Delta \geq L_{ki}^0 + L_{ij}^\Delta$: indeed L_{ij}^Δ is attained for some $\delta^* \in \Delta \subseteq \mathbb{R}_+^J$, hence $L_{kj}^\Delta \geq L_{G(p+\delta^*)}(k, i) + L_{G(p+\delta^*)}(i, j) \geq L_{ki}^0 + L_{ij}^\Delta$. Finally $z_j - z_i \geq L_{ij}^\Delta$, hence $z \in Z^{\leq M}(H)$. \square

3.3 Dominance-based linear formulation (Dom)

(AnchRob) problem is to maximize the weight of a set H such that $X^{\leq M}(H) \neq \emptyset$, or equivalently with Theorem 1, such that $Z^{\leq M}(H) \neq \emptyset$. We now introduce a new formulation derived from Theorem 1, where continuous variables z correspond to a schedule $z \in Z^{\leq M}(H)$.

$$\begin{aligned} \text{(Dom):} \quad & \max \quad \sum_{i \in J} w_i h_i \\ & \text{s.t.} \quad z_j - z_i \geq p_i && \forall (i, j) \in \mathcal{A} \\ & \quad z_t \leq M \\ & \quad z_j - z_i \geq L_{ij}^0 + (L_{ij}^\Delta - L_{ij}^0) h_j && \forall i, j \in \bar{J}, i \prec j \\ & \quad z_j \geq 0 && \forall j \in \bar{J} \\ & \quad h_j \in \{0, 1\} && \forall j \in J \end{aligned} \tag{3}$$

Proposition 2. *Formulation (Dom) is valid for (AnchRob).*

Proof. Let (z, h) be feasible for (Dom), and $H := \{i \in J : h_i = 1\}$. Note first that inequality (3) implies $z \in X^{\leq M}(H)$. Hence z is a schedule with makespan at most M and H is z -anchored. Hence $h \in \mathcal{H}$. Conversely, let $h \in \mathcal{H}$ be the incidence vector of an anchored set H . From Prop. 1 and Theorem 1, there exists $z \in Z^{\leq M}(H)$. Such a schedule z satisfies $z_j - z_i \geq L_{ij}^\Delta$ for every $i \in J \cup \{s\}$ and $j \in H$. Then z satisfies (3): indeed if $h_j = 1$ inequality (3) corresponds to inequality $z_j - z_i \geq L_{ij}^\Delta$; if $h_j = 0$ then (3) amounts to $z_j - z_i \geq L_{ij}^0$, which holds since z is a schedule of $G(p)$. \square

Note that precedence constraint $z_j - z_i \geq p_i$ associated with arc $(i, j) \in \mathcal{A}$ is implied by (3) since $i \prec j$ and $L_{ij}^0 \geq p_i$.

We now introduce a family of valid inequalities. Let $j \in J$. By inequality (3) with $i = s$, it comes that $z_j \geq L_{sj}^0 + (L_{sj}^\Delta - L_{sj}^0) h_j$. Since $z \in X^{\leq M}$ it satisfies $z_j + L_{jt}^0 \leq M$, thus leading to $M \geq L_{sj}^0 + (L_{sj}^\Delta - L_{sj}^0) h_j + L_{jt}^0$. Hence the following valid inequality

$$h_j \leq \left\lfloor \frac{M - (L_{sj}^0 + L_{jt}^0)}{L_{sj}^\Delta - L_{sj}^0} \right\rfloor. \tag{4}$$

In Section 4 they will be discussed with respect to the polyhedral characterization under box uncertainty.

3.4 (Dom) is stronger than other known formulations

In this section it is shown that formulation (Dom) is stronger than (Std) and stronger than (Lay), the previously known formulation from [3] dedicated to budgeted uncertainty.

Consider first the continuous relaxations of (Dom), (Std) and the variant of (Std) with quadratic constraint (1). It turns out that they can easily be compared. Indeed for every $h_i, h_j \in [0, 1]$, it holds that $h_j \geq h_i h_j$. Hence the right-hand side of inequality (3) is tighter than the right-hand side of inequality (1). Also for every $h_i, h_j \in [0, 1]$, it holds that $(1-h_i)(1-h_j) \geq 0$ or equivalently $h_i h_j \geq h_i + h_j - 1$. Hence the right-hand side of inequality (1) is tighter than the right-hand side of inequality (2).

Let us now investigate the special case of budgeted uncertainty, and compare (Dom) with formulation (Lay) from [3], that we now recall. Formulation (Lay) involves anchoring variables $h_j \in \{0, 1\}$ for every $j \in J$, and continuous variables x_j^γ for every $\gamma \in \{0, \dots, \Gamma\}$, $j \in \bar{J}$. Formulation (Lay) is based on a so-called *layered graph* $G^{\text{lay}}(h)$ associated with $h \in [0, 1]^J$ and built as follows. The layered graph $G^{\text{lay}}(h)$ is formed with $\Gamma + 1$ copies of the precedence graph called *layers* indexed from 0 to Γ . Let i^γ denote the copy of job i in layer γ . The layered graph features three types of arcs. Horizontal arcs are copies of arcs of $G(p)$, i.e., arcs (i^γ, j^γ) for $(i, j) \in \mathcal{A}$, with length p_i . Transversal arcs are $(i^{\gamma+1}, j^\gamma)$ for $(i, j) \in \mathcal{A}$, with length $p_i + \widehat{\delta}_i$. Vertical arcs are (i^γ, i^Γ) for $i \in J$, $\gamma < \Gamma$, with length $-D_j(1-h_j)$, where $D_j = L_{G(p+\widehat{\delta})}(s, j) - L_{G(p)}(s, j)$ for every $j \in J$. It was shown that $h \in \{0, 1\}^J$ is the incidence vector of an anchored set if and only if there exists x a schedule of $G^{\text{lay}}(h)$ such that $x_t^\Gamma \leq M$. This leads to the formulation

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

Variables x_j^Γ , $j \in \bar{J}$ from layer Γ can be thought of as a baseline schedule such that $H = \{i \in J : h_i = 1\}$ is x^Γ -anchored. Other continuous variables x_j^γ , $j \in \bar{J}$, $\gamma < \Gamma$ can be regarded as additional variables.

In order to compare linear relaxations of (Lay) and (Dom), we show how to project explicitly those two formulations on the space of h variables. Let \mathcal{C} be the set of s - t paths in the transitive closure of G . Note that there is one-to-one correspondence with chains of the poset (J, \prec) . Let \mathcal{P}^{Lay} (resp. \mathcal{P}^{Dom}) denote the polytope of solutions $((x^\gamma)_{\gamma \in \{0, \dots, \Gamma\}}, h)$ (resp. (z, h)) that are feasible for the continuous relaxation of formulation (Lay) (resp. (Dom)).

Proposition 3.

$$\text{Proj}_h(\mathcal{P}^{\text{Dom}}) = \left\{ h \in [0, 1]^J : \sum_{\substack{(i,j) \in \mathcal{C} \\ j \neq t}} (L_{ij}^0 + (L_{ij}^\Delta - L_{ij}^0)h_j) + L_{jct}^0 \leq M \forall C \in \mathcal{C} \right\}$$

$$\text{Proj}_h(\mathcal{P}^{\text{Lay}}) = \left\{ h \in [0, 1]^J : \sum_{\substack{(i,j) \in C \\ j \neq t}} (L_{ij}^\Delta - D_j(1-h_j)) + L_{j_C t}^0 \leq M \forall C \in \mathcal{C} \right\}$$

where j_C denotes the last vertex of path C before sink t .

Proof. Let $h \in [0, 1]^J$. Let $\overline{G}(h)$ denote the transitive closure of G , where arc (i, j) is given the weight $L_{ij}^0 + (L_{ij}^\Delta - L_{ij}^0)h_j$, with $h_t = 0$ for the ease of notation. Then $(z, h) \in \mathcal{P}^{\text{Dom}}$ if and only if z is a schedule of $\overline{G}(h)$ with makespan $z_t \leq M$, by definition of formulation (Dom). The existence of such z is equivalent to $L_{\overline{G}(h)}(s, t) \leq M$; or equivalently, every $s-t$ path in $\overline{G}(h)$ has length at most M . The length of path $C \in \mathcal{C}$ in $\overline{G}(h)$ is exactly the left-hand side of the proposed inequality, hence the result.

A similar proof holds for formulation (Lay). Let $h \in [0, 1]^J$. Let $\ell_{G^{\text{lay}}(h)}(R)$ denote the length of a path R in the layered graph $G^{\text{lay}}(h)$. It holds that $h \in \text{Proj}_h(\mathcal{P}^{\text{Lay}})$ if and only if there exists x a schedule of $G^{\text{lay}}(h)$ such that $x_t^\Gamma \leq M$. The existence of x is equivalent to the longest path condition: $L_{G^{\text{lay}}(h)}(s^\Gamma, t^\Gamma) \leq M$; or equivalently, $\ell_{G^{\text{lay}}(h)}(R) \leq M$ for every path R from s^Γ to t^Γ in the layered graph $G^{\text{lay}}(h)$. Let us now show that it is equivalent to $\sum_{(i,j) \in C, j \neq t} (L_{ij}^\Delta - D_j(1-h_j)) + L_{j_C t}^0 \leq M$ for every $C \in \mathcal{C}$.

Assume first $\ell_{G^{\text{lay}}(h)}(R) \leq M$ for every path R from s^Γ to t^Γ in $G^{\text{lay}}(h)$. Let $C \in \mathcal{C}$ be an $s-t$ path in the transitive closure of G . Consider an associated $s^\Gamma-t^\Gamma$ path R^* in the layered graph built as follows: for every arc $(i, j) \in C, j \neq t$, path R^* contains the subpath of length L_{ij}^Δ going from i^Γ to a copy j^γ of j , and the vertical arc (j^γ, j^Γ) ; path R^* also contains the subpath of length $L_{j_C t}^0$ going from j_C^Γ to t^Γ . Then R^* is an $s^\Gamma-t^\Gamma$ path in $G^{\text{lay}}(h)$, by assumption it has length at most M , hence $\sum_{(i,j) \in C, j \neq t} (L_{ij}^\Delta - D_j(1-h_j)) + L_{j_C t}^0 \leq M$.

Conversely assume $\sum_{(i,j) \in C, j \neq t} (L_{ij}^\Delta - D_j(1-h_j)) + L_{j_C t}^0 \leq M$ for every $C \in \mathcal{C}$. Let R be an $s^\Gamma-t^\Gamma$ path in $G^{\text{lay}}(h)$. Let C^* be the path defined the successive jobs j such that R contains a vertical arc (j^γ, j^Γ) ; then $C^* \in \mathcal{C}$. For every $(i, j) \in C^*$, let $R^{(i,j)}$ denote the subpath of R from i^Γ to a copy j^γ of job j . Since $R^{(i,j)}$ uses at most Γ transversal arcs, it comes $\ell_{G^{\text{lay}}(h)}(R^{(i,j)}) \leq L_{ij}^\Delta$. Similarly, let $R^{(j_C^*, t)}$ denote the subpath of R between j_C^{Γ} and t^Γ , then $\ell_{G^{\text{lay}}(h)}(R^{(j_C^*, t)}) \leq L_{j_C^* t}^0$. The total length of R is $\ell_{G^{\text{lay}}(h)}(R) = \sum_{(i,j) \in C^*, j \neq t} (\ell_{G^{\text{lay}}(h)}(R^{(i,j)}) - D_j(1-h_j)) + \ell_{G^{\text{lay}}(h)}(R^{(j_C^*, t)})$, thus upper-bounded by $\sum_{(i,j) \in C^*, j \neq t} (L_{ij}^\Delta - D_j(1-h_j)) + L_{j_C^* t}^0$. By assumption this at most M since $C^* \in \mathcal{C}$, hence $\ell_{G^{\text{lay}}(h)}(R) \leq M$. This proves the claimed result. \square

Proposition 4. *Formulation (Dom) is stronger than formulation (Lay), in the sense that $\text{Proj}_h(\mathcal{P}^{\text{Dom}}) \subseteq \text{Proj}_h(\mathcal{P}^{\text{Lay}})$.*

Proof. For every $i \prec j$, it holds that $L_{ij}^\Delta - D_j(1-h_j) \leq L_{ij}^0 + (L_{ij}^\Delta - L_{ij}^0)h_j$: indeed it is equivalent to $(D_j - L_{ij}^\Delta + L_{ij}^0)h_j \leq D_j - L_{ij}^\Delta + L_{ij}^0$ which is clearly satisfied for every $h_j \in [0, 1]$. Using the explicit definition of projections in Prop. 3, it comes that the inequalities defining the projection of (Dom) are tighter than the inequalities defining the projection of (Lay). \square

We now give an example where $\text{Proj}_h(\mathcal{P}^{\text{Dom}})$ is strictly included in $\text{Proj}_h(\mathcal{P}^{\text{Lay}})$. Let $J = \{1, 2, 3\}$ be a set of three jobs, let G be the path $(s, 1, 2, 3, t)$, and let $p_i = 1$ and $\widehat{\delta}_i = 1$ for every $i \in J$. Let also $\Gamma = 1$, and $M = 3$. Consider $h^* = (1, 0, \frac{1}{2})$. To see that $h \in \text{Proj}_h(\mathcal{P}^{\text{Lay}})$, consider the layered graph $G^{\text{lay}}(h^*)$, represented in Figure 1. The vector x defined by $x^1 = (0, 0, 1, 2, 3)$ in layer 1, $x^0 = (0, 0, 2, 3, 4)$ in layer 0 is also represented into brackets on the vertices in Figure 1.

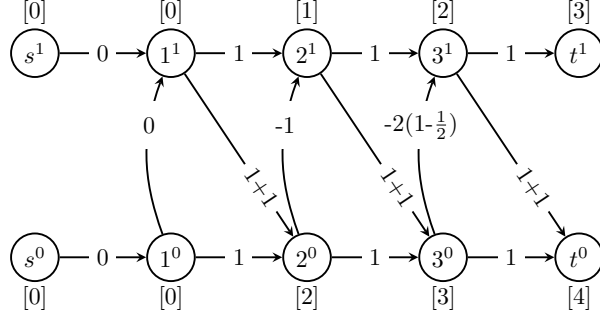


Figure 1: Layered graph $G^{\text{lay}}(h^*)$ for $h^* = (1, 0, \frac{1}{2})$.

It can be checked that vector x is a schedule of $G^{\text{lay}}(h^*)$, hence $(x, h^*) \in \mathcal{P}^{\text{Lay}}$. By contrast, vector h^* violates inequality defining $\text{Proj}_h(\mathcal{P}^{\text{Dom}})$ associated with path $P = (s, 3, t)$. Indeed $L_{s3}^0 = 2$, $L_{s3}^\Delta = 3$, $L_{3t}^0 = 1$ and the inequality is $2 + h_3 + 1 \leq 3$. Thus $h^* \notin \text{Proj}_h(\mathcal{P}^{\text{Dom}})$, and $\text{Proj}_h(\mathcal{P}^{\text{Dom}}) \subsetneq \text{Proj}_h(\mathcal{P}^{\text{Lay}})$. More precisely, it is a case where (Dom) yields a polyhedral characterization of (AnchRob), as shown next in Section 4.2.1.

4 Polyhedral characterization for special cases

In this section, we provide polyhedral characterizations of (AnchRob) for two special cases: box uncertainty and 1-disruption uncertainty for critical precedence graph.

4.1 Box uncertainty

This section is devoted to box uncertainty, which is the special case of budgeted uncertainty where $\Gamma = |J|$, and thus $\hat{\delta} \in \Delta$. In [3] a polynomial algorithm for (AnchRob) under box uncertainty was provided. This algorithm is as follows: compute \underline{x} the earliest schedule of $G(p + \hat{\delta})$ and \bar{x} the latest schedule of $G(p)$ such that $\bar{x}_t = M$; let $H^* = \{i \in J \mid \underline{x}_i \leq \bar{x}_i\}$ and $x_i = \min\{\underline{x}_i, \bar{x}_i\}$ for every $i \in \bar{J}$; return (x, H^*) .

The main result is that in this case the polytope \mathcal{Q} of anchored sets is characterized by inequalities (4).

Theorem 2. *For box uncertainty,*

$$\mathcal{Q} = \{h \in [0, 1]^J : h \text{ satisfies (4)}\}.$$

Proof. Consider schedules \underline{x} and \bar{x} , defined by $\underline{x}_j = L_{G(p+\hat{\delta})}(s, j)$ and $\bar{x}_j = M - L_{jt}^0$ for every $j \in J$. Note first that for box uncertainty, $L_{sj}^\Delta = L_{G(p+\hat{\delta})}(s, j) = \underline{x}_j$. Inequality (4) writes in this case: $h_j \leq \left\lfloor \frac{M - (L_{sj}^0 + L_{jt}^0)}{L_{sj}^\Delta - L_{sj}^0} \right\rfloor = \left\lfloor \frac{\bar{x}_j - L_{sj}^0}{\underline{x}_j - L_{sj}^0} \right\rfloor$. Equivalently, it implies $h_j \leq 1$ if $j \in H^*$, and $h_j \leq 0$ if $j \notin H^*$. Hence $\{h \in [0, 1]^J : h \text{ satisfies (4)}\} = [0, 1]^{H^*} \times \{0\}^{J \setminus H^*}$.

Every extreme point of $[0, 1]^{H^*} \times \{0\}^{J \setminus H^*}$ is the incidence vector of a set $H \subseteq H^*$. Since any subset of an anchored set is anchored, and H^* is anchored, then $[0, 1]^{H^*} \times \{0\}^{J \setminus H^*} \subseteq \mathcal{Q}$. Conversely, if H is an anchored set, its incidence vector χ^H satisfies valid inequalities (4): thus $H \subseteq H^*$. Hence $\mathcal{Q} \subseteq [0, 1]^{H^*} \times \{0\}^{J \setminus H^*}$. \square

Note that Theorem 2 also holds for any Δ that has a greatest element $\hat{\delta}$. This case is more general than Δ being a box, as mentioned in Section 2.2.

4.2 1-disruption uncertainty

In this section, set Δ is a 1-disruption uncertainty set, i.e., a budgeted uncertainty set with $\Gamma = 1$ and $\widehat{\delta}_i = \widehat{\delta}_0$ for every $i \in J$. In Section 4.2.1 the polyhedral characterization is shown for the special case with zero processing times. In Section 4.2.2 it is extended to any critical precedence graph with non-zero processing times.

4.2.1 The Unitary (AnchRob)

Assume first $p_i = 0$ for every $i \in J$. W.l.o.g. $\widehat{\delta}_0 = 1$. Finally assume deadline M integer. If M is not integer, it can be replaced with $\lfloor M \rfloor$ w.l.o.g.: indeed, since p and $\widehat{\delta}$ are integer, for any H the earliest schedule of \widetilde{G}_H is integer-valued with integer makespan.

In this special case, an instance is the precedence graph $G = (\overline{J}, \mathcal{A})$, $p = 0$, $\widehat{\delta}_0 = 1$, integer deadline M , and anchoring weights. This special case is referred to as the Unitary Anchor-Robust (U-AnchRob) problem. For unit anchoring weights, it has been identified as a polynomial case in [3] by an equivalence with a problem on posets for which a min/max theorem is known [17].

For (U-AnchRob) our main result is a characterization of the polytope through formulation (Dom). It relies on the polyhedral result of Theorem 3. Let (\overline{J}, \prec) be a poset with s (resp. t) a least (resp. greatest) element. Consider the inequalities

$$\begin{aligned}
z_i - z_s &\geq 0 & \forall i \in J & & \text{(a)} \\
z_t - z_i &\geq 0 & \forall i \in J & & \text{(b)} \\
z_t &\leq M & & & \text{(c)} \\
z_j - z_i &\geq h_j & \forall i, j \in J, i \prec j & & \text{(d)} \\
z_j &\geq 0 & \forall j \in \overline{J} & & \text{(e)} \\
h_j &\leq 1 & \forall j \in J & & \text{(f)} \\
h_j &\geq 0 & \forall j \in J & & \text{(g)}
\end{aligned}$$

and let $\mathcal{P} = \{(z, h) \in \mathbb{R}^{\overline{J}} \times \mathbb{R}^J : \text{(a)–(g)}\}$.

Theorem 3. *The polytope \mathcal{P} is integer.*

Proof. Consider the polytope \mathcal{P} formed with all pairs $(z, h) \in \mathbb{R}_+^{\overline{J}} \times \mathbb{R}_+^J$ satisfying the constraints (a)–(g). To prove integrality of \mathcal{P} , the main idea is to define an auxiliary extended polyhedron where h variables can be expressed linearly from z variables and additional z' variables. Let us define the auxiliary polyhedron \mathcal{P}' formed with all triplets $(z, z', h) \in \mathbb{R}_+^{\overline{J}} \times \mathbb{R}_+^{\overline{J}} \times \mathbb{R}^J$ satisfying the constraints

$$\begin{aligned}
z_j - z_i &\geq 0 & \forall (i, j) \in \mathcal{A} & & \text{(i)} \\
z'_j - z'_i &\geq 0 & \forall (i, j) \in \mathcal{A} & & \text{(ii)} \\
z'_j - z_i &\geq 1 & \forall (i, j) \in \mathcal{A} & & \text{(iii)} \\
z_j - z'_j &\geq -1 & \forall j \in J & & \text{(iv)} \\
z_j - z'_j &\leq 0 & \forall j \in J & & \text{(v)} \\
z_t &\leq M & & & \text{(vi)} \\
h_j &= 1 + z_j - z'_j & \forall j \in J & & \text{(vii)}
\end{aligned}$$

Let us prove the following claim. *Claim 1.* $\mathcal{P} = \text{Proj}_{z, h}(\mathcal{P}')$.

First, given $(z, h) \in \mathcal{P}$, let us prove the existence of z' such that $(z, z', h) \in \mathcal{P}'$. Define $z'_j = 1 + z_j - h_j$ for every $j \in J$, and $z'_s = z_s$, $z'_t = 1 + z_t$. Then the (in)equalities (i) to (vii) can be checked for the triplet (z, z', h) as follows.

– (i) and (vi) hold by assumption on z , and (vii) by definition of z' ;

- (ii): if $j \neq t$ then $z'_j - z'_i = z_j - z_i - h_j + h_i \geq h_i \geq 0$ by (d); if $j = t$ then $z'_j - z'_i = z_t - z_i + h_i \geq h_i \geq 0$ by (b);
- (iii): if $j \neq t$ then $z'_j - z_i = 1 + z_j - h_j - z_i \geq 1$ by (d); if $j = t$, follows from (b);
- (iv) and (v): $z_j - z'_j = -1 + h_j \in [-1, 0]$ since $h_j \in [0, 1]$.

Conversely, let $(z, z', h) \in \mathcal{P}'$. Let us check inequalities (a)–(g) for (z, h) .

- (a) holds by sum of (i) along an s – i path;
- (b) holds by sum of (i) along an i – t path;
- (c) is clear;
- (d): for $i \prec j$ we have $z_j - z_i = z_j - z'_j + z'_j - z'_k + z'_k - z_i$, where k is the last vertex distinct from j on a path from i to j in the precedence graph (possibly $k = i$). Then $(k, j) \in \mathcal{A}$ so $z'_j - z'_k \geq 1$ by (iii). Also $z'_k - z_i \geq 0$ by summing (i) along the i – k path. Hence $z_j - z_i \geq z_j - z'_j + 1 + 0 = h_j$ by (vii). Hence (d) is satisfied;
- (f) (resp. (g)) comes from (vii) and (iv) (resp. (vii) and (v)).

This completes the proof of Claim 1. \diamond

Now we prove: *Claim 2. \mathcal{P}' is integer.*

Let (z, z', h) be an extreme point of \mathcal{P}' . It satisfies the n equalities (vii) and it saturates $2(n+2)$ linearly independent inequalities among (i)–(vi). Thus (z, z') is an extreme point of $\{(z, z') \in \mathbb{R}_+^{\bar{J}} \times \mathbb{R}_+^{\bar{J}} : \text{(i)–(vi)}\}$. The constraint matrix of (i)–(vi) is totally unimodular, and the right-hand side is integer since M is integer. Hence (z, z') is integer and so is (z, z', h) . This ends the proof of Claim 2. \diamond

By Claim 1, $\mathcal{P} = \text{Proj}_{z,h}(\mathcal{P}')$. It holds that for any extreme point (z, h) of \mathcal{P} there exists z' such that (z, z', h) is an extreme point of \mathcal{P}' . By Claim 2, such (z, z', h) is integer, hence (z, h) is integer. This proves the integrality of \mathcal{P} . \square

Theorem 3 implies the following polyhedral characterization

Proposition 5. *Formulation (Dom) yields a polyhedral characterization for (U-AnchRob).*

Proof. Let us prove that the linear relaxation of (Dom) corresponds to the polytope \mathcal{P} from Theorem 3. For (U-AnchRob) for every $i, j \in \bar{J}$ such that $i \prec j$ we have $L_{ij}^0 = 0$ since $p = 0$. The worst-case longest paths values are as follows. For every $j \in J$, $L_{sj}^\Delta = 0$ if j has no predecessor except s , and $L_{sj}^\Delta = 1$ otherwise. For every $i \prec j$ with $i \neq s$, $L_{ij}^\Delta = 1$. For a pair $i \prec j$ with $i \neq s$, inequality (3) thus writes $z_j - z_i \geq h_j$, which is inequality (d) or (b) from the definition of \mathcal{P} . For a pair $s \prec j$, inequality (3) writes $z_j - z_s \geq 0$ if j has no predecessor in J , and $z_j - z_s \geq h_j$ otherwise. If j has no predecessor in J , it is inequality (a) from the definition of \mathcal{P} . Otherwise j has a predecessor $k \in J$ and inequality $z_j - z_s \geq h_j$ is dominated by inequalities $z_k - z_s \geq 0$ and $z_j - z_k \geq h_k$. Hence it is satisfied by any element of \mathcal{P} . Thus $\mathcal{P}^{\text{Dom}} = \mathcal{P}$.

By Theorem 3, \mathcal{P} is integer, hence $\text{Proj}_h(\mathcal{P})$ is integer. Namely, each extreme point of $\text{Proj}_h(\mathcal{P})$ is the incidence vector of an anchored set. Thus $\text{Proj}_h(\mathcal{P}^{\text{Dom}}) = \text{Proj}_h(\mathcal{P}) = \mathcal{Q}$, and (Dom) yields a polyhedral characterization for the problem. \square

Note that Proposition 5 implies also the integrality of schedule variables z , which are not required to be integer in general.

As a corollary, a complexity result is that Theorem 3 generalizes the polynomial case of (U-AnchRob) proven in [3] for unit anchoring weights, to any non-negative anchoring weights.

The polyhedral characterization from (Dom) can be projected out to obtain a complete description of \mathcal{Q} using h variables only. Namely, the inequalities from the projection stated in Proposition 3 are $\sum_{i \in C} h_i \leq M$ for every chain C of the subposet (J^*, \prec) , where J^* is the set of jobs with at least a predecessor different from s . It is a family of inequalities with exponential

size. In that regard, (Dom) is a compact extended formulation with additional variables z : with $O(n)$ continuous additional variables, it is possible to describe the polytope \mathcal{Q} with a polynomial $O(n^2)$ number of inequalities.

4.2.2 Critical precedence graphs

Let us now consider processing times $p \in \mathbb{R}_+^J$. Recall that the precedence graph $G(p)$ is *critical* if the length of all $s-t$ paths is the same. A case where $G(p)$ is critical is when processing times are equal to zero, since the length of all $s-t$ paths is zero. For non-zero processing times, it can also be proven that for some precedence graphs, if $G(p)$ is quasi-critical then it is critical. This holds for series-parallel precedence graphs; the result is proven in A and used for numerical results in Section 5 to generate critical precedence graphs.

Let us now consider a critical precedence graph $G(p)$, and extend the polyhedral characterization result obtained in Section 4.2.1. It is assumed that $M = L_{st}^0 + \widehat{\delta}_0 M'$ with M' integer. Indeed M can be tightened to $L_{st}^0 + \widehat{\delta}_0 \left\lfloor (M - L_{st}^0) / \widehat{\delta}_0 \right\rfloor$ w.l.o.g. Under this assumption we prove that

Theorem 4. *For 1-disruption uncertainty and critical precedence graph, formulation (Dom) yields a polyhedral characterization of (AnchRob).*

Proof. Let \mathcal{I} denote an instance of (AnchRob) for 1-disruption uncertainty with deviation $\widehat{\delta}_0$, critical precedence graph $G(p)$, and deadline M . Let z^* be the earliest schedule of $G(p)$, namely $z_i^* = L_{si}^0$ for every $i \in \overline{J}$. Since $G(p)$ is critical, for every $i < j$ it holds that $L_{si}^0 + L_{ij}^0 + L_{jt}^0 = L_{sj}^0 + L_{jt}^0$, hence $L_{ij}^0 = z_j^* - z_i^*$. Also $L_{ij}^\Delta = z_j^* - z_i^* + \widehat{\delta}_0$. Inequalities (3) then write $z_j - z_i \geq (z_j^* - z_i^*) + \widehat{\delta}_0 h_j$, or equivalently $\frac{z_j - z_i}{\widehat{\delta}_0} - \frac{z_j^* - z_i^*}{\widehat{\delta}_0} \geq h_j$. Similarly the deadline constraint is equivalent to $\frac{z_t - z_i^*}{\widehat{\delta}_0} \leq \frac{M - z_i^*}{\widehat{\delta}_0}$. Consider a new instance \mathcal{I}' of (U-AnchRob) defined by: the precedence graph G , zero processing times, and $M' = \frac{M - L_{st}^0}{\widehat{\delta}_0}$. Then M' is integer by assumption on M . Solution (z, h) is feasible for (Dom) in instance \mathcal{I} if and only if solution $(\frac{z - z^*}{\widehat{\delta}_0}, h)$ is feasible for (Dom) in instance \mathcal{I}' . Hence if (z, h) is an extreme point of \mathcal{P}^{Dom} then $(\frac{z - z^*}{\widehat{\delta}_0}, h)$ is extreme for the (U-AnchRob) instance. By Proposition 5, vector h is integer. Hence the claimed result. \square

Theorem 4 thus yields a polyhedral characterization of polynomial size for this special case. We mention that this is a polynomial case of (AnchRob) that was not identified in [3].

5 Numerical results

We investigate the impact of theoretical polyhedral results from Section 4 on the performance of formulations (Lay), (Std), (Dom) for various instance classes. In Section 5.1, instances and settings are presented. Section 5.2 is dedicated to budgeted uncertainty, and Section 5.3 to partition-budgeted and mixed-budgeted uncertainty sets.

5.1 Instances and settings

Instances are randomly generated as follows. We consider instance classes with a four-field label F1_F2_F3_F4:

- Field **F1** concerns precedence graph G
 - **ER**: precedence graphs randomly generated according to Erdos-Renyi model, i.e., arc (i, j) is in G with probability $pr = 10/n$.
 - **SP**: Series-Parallel precedence graphs, inductively generated by drawing randomly series or parallel compositions.
- Field **F2** denotes processing times p
 - **pZero**: $p_i = 0$ for every $i \in J$;
 - **pRand**: p_i is randomly generated in range $[5, 20]$;
 - **pQCri**: p is obtained by applying the following procedure: start from the values generated for class **pRand**; increase the processing time of a randomly selected job until every job is on a critical path. Hence $G(p)$ is quasi-critical.
- Field **F3** denotes deviation $\widehat{\delta}$
 - **dRand**: for instances **pRand** and **pQCri**, $\widehat{\delta}_i$ is randomly generated in $[1, \frac{1}{2}p_i]$ for every $i \in J$; for instances **pZero**, $\widehat{\delta}$ is equal to the values generated for instances **pQCri**;
 - **dUnif**: $\widehat{\delta}_i = \widehat{\delta}_0$ for every $i \in J$. Value $\widehat{\delta}_0$ is randomly selected in the deviation values of instances **dRand**.
- Field **F4** denotes the uncertainty set
 - **$\Gamma 1, \Gamma 2, \Gamma 3$** correspond to budgeted uncertainty with deviation defined by **F3** and $\Gamma = 1, 2, 3$ respectively.
 - **Partition**: jobs are partitioned into two subsets J^1 and J^2 , every job being in J^1 with probability 0.75. Given the deviation $\widehat{\delta}$ defined by **F3**, the deviation vector of the **Partition** instance is $[0.1\widehat{\delta}_i]$ for every $i \in J^1$, and $\widehat{\delta}_i$ for every $i \in J^2$. Budgets are $\Gamma^1 = 10$ and $\Gamma^2 = 1$.
 - **Mixed**: $\Delta = \Delta^1 \cup \Delta^2$ where Δ^1 has deviation $\widehat{\delta}$ defined by **F3** and $\Gamma^1 = 1$, and Δ^2 has deviation $\tau\widehat{\delta}$, with $\tau = 0.2$ and $\widehat{\delta}$ defined by **F3** and $\Gamma^2 = 10$.

Each label corresponds to a class of 10 instances. For completeness, a formal definition of series-parallel precedence graphs is given in A. It is also proven in the Appendix that instance classes **SP-pQCri** yield critical precedence graphs.

The number of jobs is set to $n = 300$. Anchoring weights are unitary. Deadline M is set to $M^{\frac{1}{2}} = \frac{1}{2}(L_{G(p)}(s, t) + L_{G(p+\widehat{\delta})}(s, t))$, that is, it is halfway between the min makespan of any schedule of $G(p)$ and the min makespan of a robust-static schedule. Unreported results showed that choosing a deadline other than $M^{\frac{1}{2}}$ leads to similar results in terms of the comparison of formulations. For budgeted uncertainty, the budget is $\Gamma \in \{1, 2, 3\}$. The choice of a small uncertainty budget was previously motivated in the literature, see, e.g., [3, 13].

For each instance, formulations (**Dom**), (**Std**), and (**Lay**) have been implemented using Julia 0.6.2, JuMP 0.18.5. Mixed-integer programs are solved with CPLEX 12.8 on a PC under Windows 10 with Intel Core i7-7500U CPU 2.90GHz and 8 Go RAM. The time limit is 300 seconds.

The valid inequalities (4) appear to be added by CPLEX on the fly. Thus they are not hardcoded in formulations.

5.2 Impact of instance parameters for budgeted uncertainty

Let us first investigate the case of budgeted uncertainty. Table 1 and Table 2 present the results for **ER** and **SP** instances respectively. Each table presents results relative to 8 instance classes: the first 6 instance classes are with $\Gamma = 1$ and with all combinations of processing times and deviations, and the last 2 instance classes are with $\Gamma = 2$ and $\Gamma = 3$. For each instance class, checkmarks in the first three columns indicate if the assumptions of Theorem 4 are matched:

- crit.: the precedence graph $G(p)$ is critical;
- unif.: deviation $\hat{\delta}$ is uniform;
- $\Gamma 1$: $\Gamma = 1$.

The tables feature:

- opt: average optimal value for instances solved optimally;
- for each formulation (Lay), (Std) and (Dom):
 - #solved: number of instances, out of 10, solved optimally within the time limit;
 - gap: average final gap of unsolved instances;
 - time: average computation time for solved instances in seconds;
 - LPGap: average gap $\frac{b-opt}{opt}$ between integer optimum opt and linear bound b ;
 - CPXGap: average gap obtained by CPLEX at root node.

The computation times do not include the preprocessing time for computing L_{ij}^{Δ} values. The computation is done by a dynamic programming algorithm linear in $\Gamma|\mathcal{A}|$. Its running time is negligible with respect to MIP computation time: on average 0.153 seconds for **ER** instances and 0.161 seconds for **SP** instances.

Let us now comment on the impact of the instance parameters.

Polyhedral characterization cases. Instance classes **ER_pZero_dUnif_Γ1**, **SP_pZero_dUnif_Γ1**, and **SP_pQCri_dUnif_Γ1**, correspond to polyhedral characterization cases. As expected, formulation (Dom) solves the problem in less than one second and LPGap = 0%. Formulation (Lay) has non-zero LPGap, but CPLEX adds suitable cuts to close the gap at root node.

Impact of uniform deviation. Consider now the 6 instance classes **F1_F2_dUnif_Γ1**. These are the first three row entries of Table 1 and the first three row entries of Table 2. On these 6 instance classes, formulations (Dom) and (Lay) still behave well. (Dom) solves all instances in less than one second. In particular, its LPGap is still very small: at most 0.42%. (Lay) also performs well, the LPGap of (Lay) is larger, but CPXGap is comparable for (Dom) and (Lay). We note that uniform deviation has an important impact on the performance of formulations. Consider, e.g., instance classes **SP_pZero_dUnif_Γ1** and **SP_pZero_dRand_Γ1**. For uniform deviation (Dom) is integer ; for non-uniform deviation it has 9.70% CPXGap and solves only 4 instances out of 10 within the time limit. By contrast with (Dom) and (Lay) on these instances, formulation (Std) performs very poorly and solves only 41 instances out of 120 (vs. 107 out of 120 for (Dom)).

Impact of the precedence graph. The impact of the precedence graph being critical is limited, as shown for example by the comparison of instances **ER_pZero** and **ER_pQCri**. Both are efficiently solved, while the precedence graph is critical for the former, and not critical for the latter. Even more, instances with **pRand** appear to be easy instances, while they do not have critical precedence graphs. An interpretation is that for such instances, a large number of jobs are not on critical paths and thus they can be anchored; note, e.g., that the optimal value is greater for **pRand** instances than for others.

Impact of uncertainty budget. When Γ is increased, the performance of (Lay) deteriorates. It gets even worse than (Std) for **ER** instances, see **ER_pZero_dUnif_Γ3** where (Std) solves 3

instances and (Lay) solves 1 instance. Importantly, the size of formulation (Lay) increases with Γ . For (Dom) and (Std) only the values of the coefficients L_{ij}^{Δ} depend on the budget, and not the size of the formulation.

instance	crit.	unif.	$\Gamma 1$	opt		#solved	gap	time(s)	LPGap	CPXGap
ER_pZero.dUnif. $\Gamma 1$	✓	✓	✓	271.90	(Lay)	10	-	1	9.20%	0.90%
					(Std)	1	1.19%	144	5.14%	3.07%
					(Dom)	10	-	<1	0%	0%
ER_pQCri.dUnif. $\Gamma 1$		✓	✓	274.30	(Lay)	10	-	26	8.39%	0.87%
					(Std)	0	2.05%	-	8.04%	4.29%
					(Dom)	10	-	1	0.42%	0.24%
ER_pRand.dUnif. $\Gamma 1$		✓	✓	290.40	(Lay)	10	-	<1	3.06%	0.06%
					(Std)	10	-	30	3.02%	1.62%
					(Dom)	10	-	<1	0.10%	0%
ER_pZero.dRand. $\Gamma 1$	✓		✓	214.77	(Lay)	9	2.84%	21	34.15%	7.93%
					(Std)	9	4.92%	65	28.53%	4.68%
					(Dom)	9	1.81%	11	21.13%	3.28%
ER_pQCri.dRand. $\Gamma 1$			✓	225.80	(Lay)	8	0.98%	21	28.54%	7.70%
					(Std)	6	2.30%	41	30.25%	5.41%
					(Dom)	10	-	31	18.09%	3.69%
ER_pRand.dRand. $\Gamma 1$			✓	290.20	(Lay)	10	-	<1	3.00%	0.36%
					(Std)	10	-	<1	3.02%	1.74%
					(Dom)	10	-	<1	0.82%	0.07%
ER_pZero.dUnif. $\Gamma 2$	✓	✓		255.50	(Lay)	2	1.88%	113	15.11%	9.93%
					(Std)	1	4.20%	237	10.93%	7.58%
					(Dom)	10	-	53	3.91%	2.46%
ER_pZero.dUnif. $\Gamma 3$	✓	✓		243.50	(Lay)	1	4.41%	111	19.77%	13.69%
					(Std)	3	5.48%	153	15.76%	7.78%
					(Dom)	10	-	105	7.18%	3.51%

Table 1: ER instances, budgeted uncertainty

instance	crit.	unif.	$\Gamma 1$	opt		#solved	gap	time(s)	LPGap	CPXGap
SP_pZero.dUnif. $\Gamma 1$	✓	✓	✓	255.90	(Lay)	10	-	<1	17.09%	0%
					(Std)	0	3.34%	-	8.64%	6.81%
					(Dom)	10	-	<1	0%	0%
SP_pQCri.dUnif. $\Gamma 1$	✓	✓	✓	255.90	(Lay)	10	-	<1	17.09%	0%
					(Std)	0	6.88%	-	14.82%	8.96%
					(Dom)	10	-	<1	0%	0%
SP_pRand.dUnif. $\Gamma 1$		✓	✓	262.80	(Lay)	10	-	<1	14.19%	0.42%
					(Std)	0	6.86%	-	13.36%	7.29%
					(Dom)	10	-	<1	0.05%	0%
SP_pZero.dRand. $\Gamma 1$	✓		✓	247.00	(Lay)	4	3.72%	3	32.46%	16.43%
					(Std)	2	18.60%	11	25.97%	20.07%
					(Dom)	4	9.12%	15	16.42%	9.70%
SP_pQCri.dRand. $\Gamma 1$	✓		✓	247.00	(Lay)	4	3.52%	3	32.38%	16.72%
					(Std)	2	23.58%	30	35.21%	27.94%
					(Dom)	4	9.83%	14	16.54%	9.90%
SP_pRand.dRand. $\Gamma 1$			✓	268.60	(Lay)	10	-	3	11.51%	6.97%
					(Std)	1	6.09%	193	11.05%	9.34%
					(Dom)	10	-	16	2.26%	0.92%
SP_pQCri.dUnif. $\Gamma 2$	✓	✓		246.66	(Lay)	7	0.85%	45	22.65%	10.58%
					(Std)	0	11.80%	-	20.92%	16.63%
					(Dom)	9	0.77%	31	2.44%	1.35%
SP_pQCri.dUnif. $\Gamma 3$	✓	✓		246.14	(Lay)	5	2.77%	57	26.40%	16.38%
					(Std)	1	17.42%	215	25.47%	20.68%
					(Dom)	7	2.19%	38	3.92%	3.29%

Table 2: SP instances, budgeted uncertainty

5.3 Beyond budgeted uncertainty

Let us now present results when Δ is an uncertainty set with several budgets.

5.3.1 Partition-budgeted uncertainty set

Table 3 and Table 4 give computational results for ER and SP instances under **Partition** uncertainty. In this case, the L_{ij}^Δ values were computed by a dynamic programming algorithm with complexity linear in $\Gamma^1\Gamma^2|\mathcal{A}|$. The computation was done in 4.364 seconds on average for ER instances, and 4.916 seconds on average for SP instances.

It comes that (Dom) solves all 60 ER instances, and 54 SP instances, which is better than for budgeted uncertainty. For SP instances, the results are comparable to those of budgeted uncertainty. By contrast, ER instances appear to be very easy for this uncertainty set. Hence solving the MIP formulation (Dom) for **Partition** uncertainty does not seem harder than for budget Γ_1 . This highlight that (Dom) can be readily used to handle several budget constraints.

instance	crit.	unif.	Γ_1	opt		#solved	gap	time(s)	LPGap	CPXGap
ER_pZero.dUnif.Partition	✓	✓		286.50	(Lay)	-	-	-	-	-
					(Std)	10	-	<1	2.17%	0.07%
					(Dom)	10	-	<1	0%	0%
ER_pQCri.dUnif.Partition		✓		284.30	(Lay)	-	-	-	-	-
					(Std)	10	-	1	4.92%	0.22%
					(Dom)	10	-	<1	1.53%	0%
ER_pRand.dUnif.Partition		✓		294.50	(Lay)	-	-	-	-	-
					(Std)	10	-	<1	1.72%	0.05%
					(Dom)	10	-	<1	0.34%	0%
ER_pZero.dRand.Partition	✓			246.10	(Lay)	-	-	-	-	-
					(Std)	10	-	<1	22.03%	0%
					(Dom)	10	-	<1	20.22%	0%
ER_pQCri.dRand.Partition				260.70	(Lay)	-	-	-	-	-
					(Std)	10	-	<1	14.45%	0%
					(Dom)	10	-	<1	12.13%	0%
ER_pRand.dRand.Partition				293.10	(Lay)	-	-	-	-	-
					(Std)	10	-	<1	2.14%	0%
					(Dom)	10	-	<1	0.77%	0%

Table 3: ER instances, **Partition** uncertainty

instance	crit.	unif.	$\Gamma 1$	opt		#solved	gap	time(s)	LPGap	CPXGap
SP_pZero.dUnif.Partition	✓	✓		274.70	(Lay)	-	-	-	-	-
					(Std)	8	1.36%	14	4.48%	2.01%
					(Dom)	10	-	<1	0%	0%
SP_pQCri.dUnif.Partition	✓	✓		273.44	(Lay)	-	-	-	-	-
					(Std)	4	7.42%	43	10.57%	8.29%
					(Dom)	9	3.22%	14	1.43%	0.84%
SP_pRand.dUnif.Partition		✓		284.90	(Lay)	-	-	-	-	-
					(Std)	8	2.12%	57	4.89%	2.71%
					(Dom)	10	-	<1	0.17%	0%
SP_pZero.dRand.Partition	✓			246.87	(Lay)	-	-	-	-	-
					(Std)	7	11.40%	1	23.12%	4.54%
					(Dom)	8	4.05%	33	19.43%	2.79%
SP_pQCri.dRand.Partition	✓			247.42	(Lay)	-	-	-	-	-
					(Std)	7	13.46%	1	27.36%	5.16%
					(Dom)	7	3.28%	<1	19.33%	2.66%
SP_pRand.dRand.Partition				279.10	(Lay)	-	-	-	-	-
					(Std)	7	4.89%	44	7.10%	5.05%
					(Dom)	10	-	1	1.85%	0.41%

Table 4: SP instances, Partition uncertainty

5.3.2 Mixed-budgeted uncertainty sets

Table 5 and Table 6 give computational results for ER and SP instances under Mixed uncertainty. In this case $\Delta = \Delta^1 \cup \Delta^2$ where Δ^1 is the uncertainty set corresponding to instance classes with fourth field $\Gamma 1$. The L_{ij}^{Δ} values were precomputed by the same dynamic programming algorithm as for budgeted uncertainty. This computation was done in 0.152 seconds on average for ER instances, and 0.154 seconds on average for SP instances.

First, (Dom) solves optimally 59 ER instances and 42 SP instances for Mixed, in comparison with 59 ER instances and 48 SP instances for budgeted uncertainty ($\Gamma = 1$). That is, the performance of the formulation is not very sensitive to the change of uncertainty set.

Some conclusions given in Section 5.2 also hold for Mixed uncertainty. Namely, instances with dUnif are easier than instances with dRand. It can be related to the influence of uniform deviation on the performance of (Dom), in connection with the polyhedral characterization result.

The optimal number of anchored jobs is often the same for Δ or Δ^1 . Namely, the value of opt can be compared between Table 1 and Table 5 for ER instances, and between Table 2 and Table 6 for SP instances. For example for instance classes SP_pZero.dRand. $\Gamma 1$ and SP_pZero.dRand.Mixed the average optimal value is equal to 247.00, hence all instances have the same optimal value for uncertainty set Δ and Δ^1 . This means that the uncertainty set can be extended from Δ^1 to $\Delta = \Delta^1 \cup \Delta^2$ without deteriorating the number of anchored jobs.

instance	crit.	unif.	Γ 1	opt		#solved	gap	time(s)	LPGap	CPXGap
ER_pZero.dUnif.Mixed	✓	✓		271.90	(Lay)	-	-	-	-	-
					(Std)	1	1.17%	286	5.14%	3.06%
					(Dom)	10	-	<1	0%	0%
ER_pQCri.dUnif.Mixed		✓		274.20	(Lay)	-	-	-	-	-
					(Std)	0	2.13%	-	8.08%	4.28%
					(Dom)	10	-	1	0.45%	0.24%
ER_pRand.dUnif.Mixed		✓		290.40	(Lay)	-	-	-	-	-
					(Std)	10	-	26	3.02%	1.62%
					(Dom)	10	-	<1	0.10%	0.03%
ER_pZero.dRand.Mixed	✓			214.77	(Lay)	-	-	-	-	-
					(Std)	9	5.35%	66	28.53%	4.61%
					(Dom)	9	1.83%	13	21.13%	3.22%
ER_pQCri.dRand.Mixed				225.80	(Lay)	-	-	-	-	-
					(Std)	7	2.61%	77	30.19%	5.09%
					(Dom)	10	-	22	18.09%	3.49%
ER_pRand.dRand.Mixed				290.20	(Lay)	-	-	-	-	-
					(Std)	10	-	<1	3.02%	1.78%
					(Dom)	10	-	<1	0.82%	0.07%

Table 5: ER instances, Mixed uncertainty

instance	crit.	unif.	Γ 1	opt		#solved	gap	time(s)	LPGap	CPXGap
SP_pZero.dUnif.Mixed	✓	✓		255.70	(Lay)	-	-	-	-	-
					(Std)	0	3.60%	-	8.72%	6.84%
					(Dom)	10	-	1	0.06%	0%
SP_pQCri.dUnif.Mixed	✓	✓		246.75	(Lay)	-	-	-	-	-
					(Std)	2	24.10%	37	35.77%	24.48%
					(Dom)	4	10.15%	13	16.59%	9.99%
SP_pRand.dUnif.Mixed		✓		262.80	(Lay)	-	-	-	-	-
					(Std)	0	6.74%	-	13.36%	7.64%
					(Dom)	10	-	1	0.05%	0%
SP_pZero.dRand.Mixed	✓			247.00	(Lay)	-	-	-	-	-
					(Std)	2	19.25%	16	26.63%	19.47%
					(Dom)	4	9.43%	13	16.40%	10.19%
SP_pQCri.dRand.Mixed	✓			247.00	(Lay)	-	-	-	-	-
					(Std)	2	23.86%	30	35.52%	25.37%
					(Dom)	4	9.15%	12	16.26%	10.19%
SP_pRand.dRand.Mixed				268.40	(Lay)	-	-	-	-	-
					(Std)	1	6.03%	179	11.00%	9.48%
					(Dom)	10	-	26	2.34%	1.08%

Table 6: SP instances, Mixed uncertainty

5.4 Conclusion on numerical experiments

In the numerical experiments, we evaluated the performance of formulations for budgeted uncertainty sets, and uncertainty sets obtained by union or intersection of budgeted uncertainty sets. For budgeted uncertainty sets, numerical tests showed that (Dom) outperforms the previously known formulation (Lay), that was dedicated to budgeted uncertainty. An advantage of formulation (Dom) over (Lay) is that the size of (Dom) is independent of the budget Γ , while (Lay) has $O(n\Gamma)$ variables.

We then investigated the impact of the parameters on the performance of (Dom). The influence of uniform deviation and small uncertainty budget is important, while the impact of critical precedence graphs is not significant on the efficiency of the formulation. Interestingly (Dom) is efficient for instances that are not matching the polyhedral characterization case, but where deviation $\hat{\delta}$ is uniform and Γ is small.

The proposed approach is to precompute the L_{ij}^Δ values, then solve the obtained MIP formulation. This allowed us to solve the problem for a variety of uncertainty sets for which no linear formulation was previously investigated. The precomputing time remains small (at most 5 seconds for partition-budgeted uncertainty sets) on the considered instances, and the MIP computation time for (Dom) is comparable to that under budgeted uncertainty. While it was expected that (Dom) would outperform standard linearization (Std), the computational interest of applying the dominance is highlighted by the number of solved instances: on a total number of 400 instances, 358 are solved by (Dom), only 191 by (Std).

6 Conclusion

In the present work we investigated a versatile mixed-integer programming approach for the (AnchRob) problem. This led to a linear formulation that is applicable to any uncertainty set, provided that an algorithm for precomputing the worst-case longest paths values is available. This widens the range of uncertainty sets for which MIP formulations for (AnchRob) are known. The keypoint for establishing a strong MIP formulation is the analysis of the combinatorial properties of (AnchRob), among which a dominance property. This property allows for a characterization of the anchored sets polytope in interesting special cases. The theoretical positive results for the dominance-based formulation also go together with good numerical performances around the polyhedral characterization case, for both budgeted uncertainty and uncertainty sets with several budgets.

An interesting research direction is to benefit from the obtained polyhedral results to solve large-scale instances. Another direction is to extend these results and tackle project scheduling under resource constraints. Resource constraints are indeed very salient in applications. It is worth investigating how anchored jobs may interfere with resource constraints, and see whether dominance may help again in the design of efficient linear formulations. Another perspective is to identify other problems where an anchor-robust counterpart could be defined, and solved efficiently with mixed-integer programming.

A Series-parallel precedence graphs

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. We assume that G_1 and G_2 have strictly more than 2 vertices.

Series-parallel precedence graphs are series-parallel digraphs with terminals the dummy jobs s and t .

Proposition 6. *If G is series-parallel and $G(p)$ is quasi-critical, then $G(p)$ is critical.*

Proof. If G is a path, then $G(p)$ is critical. If G is obtained by series composition of G^1 and G^2 ; then any $s-t$ path is formed by an s^1-t^1 path in G^1 and an s^2-t^2 path in G^2 . Hence if G^1 and G^2 are critical, it follows that G is critical. If G is obtained by parallel composition of

G^1 and G^2 , both critical, let $i \in G^1$ and $j \in G^2$. Then every s - t path going through G^1 (resp. G^2) has length $L_{G^1}(s, t) = L_{G^1}(s, i) + L_{G^1}(i, t)$ (resp. $L_{G^2}(s, t) = L_{G^2}(s, j) + L_{G^2}(j, t)$). If G is quasi-critical, $L_G(s, i) + L_G(i, t) = L_G(s, j) + L_G(j, t)$, and it follows that all s - t paths of G have same length. \square

References

- [1] Josette Ayoub and Michael Poss. Decomposition for adjustable robust linear optimization subject to uncertainty polytope. *Computational Management Science*, 13(2):219–239, 2016.
- [2] Aharon Ben-Tal, A. P. Goryashko, E. Guslitzer, and Arkadi Nemirovski. Adjustable robust solutions of uncertain linear programs. *Mathematical Programming*, 99(2):351–376, 2004.
- [3] Pascale Bendotti, Philippe Chrétienne, Pierre Fouilhoux, and Adèle Pass-Lanneau. The Anchor-Robust Project Scheduling Problem. May 2019.
- [4] Pascale Bendotti, Philippe Chrétienne, Pierre Fouilhoux, and Adèle Pass-Lanneau. Anchored rescheduling problems under generalized precedence constraints. In Mourad Baiou, Bernard Gendron, Oktay Günlük, and Ali Ridha Mahjoub, editors, *Combinatorial Optimization. ISCO 2020*, volume 12176 of *Lecture Notes in Computer Science*, 2020.
- [5] 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(1):67–74, 2017.
- [6] Dimitris Bertsimas and Constantine Caramanis. Finite adaptability in multistage linear optimization. *IEEE Transactions on Automatic Control*, 55(12):2751–2766, 2010.
- [7] Dimitris Bertsimas and Melvyn Sim. The price of robustness. *Operations Research*, 52:35–53, 2004.
- [8] Alain Billionnet, Marie-Christine Costa, and Pierre-Louis Poirion. 2-stage robust MILP with continuous recourse variables. *Discrete Applied Mathematics*, 170:21–32, 2014.
- [9] Christoph Buchheim and Jannis Kurtz. Robust combinatorial optimization under convex and discrete cost uncertainty. *EURO Journal on Computational Optimization*, 6(3):211–238, 2018.
- [10] 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.
- [11] Virginie Gabrel, Cécile Murat, and Aurélie Thiele. Recent advances in robust optimization: An overview. *European Journal of Operational Research*, 235(3):471–483, 2014.
- [12] Willy Herroelen and Roel Leus. Project scheduling under uncertainty: Survey and research potentials. *European Journal of Operational Research*, 165:289–306, 2002.
- [13] Willy Herroelen and Roel Leus. The construction of stable project baseline schedules. *European Journal of Operational Research*, 156(3):550–565, 2004.
- [14] 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.

- [15] Michel Minoux. Robust linear programming with right-hand-side uncertainty, duality and applications. In *Encyclopedia of Optimization, Second Edition*, pages 3317–3327. 2009.
- [16] M. Pinedo. *Scheduling: Theory, Algorithms, and Systems*. Springer, Boston, MA, 4th edition, 2002.
- [17] Alexander Schrijver. *Combinatorial Optimization – Polyhedra and Efficiency*. Springer-Verlag Berlin Heidelberg, 2003.
- [18] Onur Şeref, Ravindra K. Ahuja, and James B. Orlin. Incremental network optimization: Theory and algorithms. *Operations Research*, 57(3):586–594, 2009.
- [19] Allen L. Soyster. Technical note – convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations Research*, 21(5):1154–1157, 1973.
- [20] Bo Zeng and Long Zhao. Solving two-stage robust optimization problems using a column-and-constraint generation method. *Operations Research Letters*, 41(5):457 – 461, 2013.