Mixed-Integer Programming Formulations for the Anchor-Robust Project Scheduling Problem

Adèle Pass-Lanneau[1,2], Pascale Bendotti[1,2], Philippe Chrétienne[2] and Pierre Fouilhoux[2]

[1] EDF R&D, Palaiseau, France
{adele.pass-lanneau, pascale.bendotti}@edf.fr
[2] Sorbonne Université, CNRS, LIP6, Paris, France
{philippe.chretienne, pierre.fouilhoux}@lip6.fr

Keywords: project scheduling, robust 2-stage optimization, anchored decisions, mixed-integer programming.

1 Introduction

In project management, computing a schedule ahead of time is often necessary to ensure the availability of equipment or staff. When the schedule is about to start, the project data may have changed, e.g., some jobs may be longer than expected in the first place. Hence the schedule computed previously has to be modified accordingly. However, in the context of an industrial complex project, some jobs might be difficult or costly to reschedule. The decision maker thus needs a guarantee over the starting times of these jobs when choosing an initial schedule.

The present work follows the general framework of (Bendotti *et al.* 2017), (Bendotti *et al.* 2019) to integrate such a criterion. We consider a project scheduling problem where jobs must be scheduled while respecting precedence constraints. Processing times of jobs have a nominal value but they may deviate from it by an uncertain amount, which is supposed to be lying in an *uncertainty set*. A *baseline schedule* is a schedule of the instance with nominal processing times that satisfies a given global deadline M. Given a baseline schedule x, subsets of *anchored jobs* are defined as subsets whose starting times in x can be guaranteed whatever the realization of processing times in the uncertainty set. Each job is associated with an anchoring weight, and the Anchor-Robust Project Scheduling Problem (AnchRob) is to find a baseline schedule x and an anchored subset H, so that the total weight of H is maximized.

The (AnchRob) problem was introduced in (Bendotti *et al.* 2019) as a tool for achieving a trade-off between robust-static project scheduling – where a schedule is computed for the worst-case processing times – and the adaptive robust project scheduling problem studied in (Minoux 2008) – where a makespan is computed, but no baseline schedule. (AnchRob) produces a baseline schedule whose makespan is controlled by the deadline M, and the adaptiveness of the solution is controlled through anchored jobs. (AnchRob) was studied in the case of budgeted uncertainty, where at most Γ processing times may deviate from their nominal values at a time. It was proven NP-hard and a MIP formulation was provided: it has a polynomial number of variables and constraints, but it has a poor linear relaxation value due to "bigM" values. Instances up to 200 jobs were solved in up to 4 minutes using this formulation.

Contributions. The present work investigates improved MIP formulations for (AnchRob). We introduce two MIP formulations, that are valid for a wider class of uncertainty sets beyond budgeted uncertainty. Two types of decisions variables are considered: for each job $i \in J$ an anchoring variable $h_i \in \{0,1\}$ to indicate whether job i is anchored, for each job $i \in \bar{J}$ a continuous variable $z_i \geq 0$ for the starting time of job i in the baseline schedule. The first formulation $\mathcal{F}_{zh}$ uses both types of variables, and it has polynomial size.

The second formulation $\mathcal{F}_h$ uses only anchoring variables. It has an exponential number of constraints for which separation algorithms are proposed. We also prove that $\mathcal{F}_{zh}$ can be projected explicitely on anchoring variables, thus yielding an alternative formulation in anchoring variables only. We establish that formulations $\mathcal{F}_{zh}$ and $\mathcal{F}_h$ are not comparable, and that they both give a better bound than the MIP formulation from (Bendotti *et al.* 2019). From this analysis, we design a customized MIP formulation that takes advantage of $\mathcal{F}_h$ and $\mathcal{F}_{zh}$. In particular polyhedral properties are used to improve the separation of inequalities from $\mathcal{F}_h$ and devise an efficient Branch-and-Cut algorithm.

For the sake of brevity all proofs are omitted.

2 The Anchor-Robust Project Scheduling Problem

Let us define more formally the Anchor-Robust Project Scheduling Problem. A set of jobs J 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 source (resp. sink) 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 $G(p)$. We consider a 2-stage framework where the processing times of jobs are uncertain. In first stage, the decision maker has an instance $G(p)$ to solve, called *baseline instance*, where p is the nominal value of processing times. The project is given a deadline $M \geq 0$. A *baseline schedule* is a schedule of $G(p)$ with makespan at most M. In second stage, the instance will be some $G(p+\delta)$, where the vector $\delta \in \mathbb{R}_+^J$ is a *disruption*. In general a baseline schedule will not remain feasible in second stage. As done classically in robust optimization, we consider that the decision maker wants to hedge against a collection $(G(p+\delta))_{\delta \in \Delta}$ of second-stage instances, where $\Delta \subseteq \mathbb{R}_+^J$ is the *uncertainty set*. Given a baseline schedule x^0, a subset H of jobs is *anchored* with respect to x^0 if for every $\delta \in \Delta$, the baseline schedule can be repaired into a feasible solution without changing starting times of jobs in H, 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 $i \in H$. Finally, each job $i \in J$ is associated with an *anchoring weight* $a_i \in \mathbb{R}_+$. (AnchRob) is then to find a baseline schedule x^0 and subset of jobs H anchored w.r.t. x^0, so that the total weight of anchored jobs $\sum_{i \in H} a_i$ is maximized.

Let us now recall a characterization of anchored jobs from (Bendotti *et al.* 2019) that will be used in the sequel. Given $i, j \in \bar{J}$ and $\delta \in \mathbb{R}_+^J$ let $L_\delta(i,j)$ denote the length of the longest i–j path in $G(p+\delta)$ (and $-\infty$ if there is no such path). In particular, $L_0(i,j)$ is the length of the longest i–j path in $G(p)$. Let $L_\Delta(i,j)$ denote $\max_{\delta \in \Delta} L_\delta(i,j)$.

Proposition 1. *(Bendotti* et al. *2019) Let x be a schedule of $G(p)$. A set H is anchored w.r.t. x iff $x_j - x_i \geq L_\Delta(i,j)$ for every $i \in H \cup \{s\}$, $j \in H$.*

In this work, the set Δ is supposed to be a subset of $\mathbb{R}_+^J$ such that: for every $\delta \in \Delta$, for every $J' \subseteq J$, the vector δ' defined by $\delta'_i = \delta_i$ if $i \in J'$, and $\delta_i = 0$ otherwise is still an element of Δ. We consider that the values $L_\Delta(i,j)$ for every $i, j \in \bar{J}$ are known and given as input. No other information on the uncertainty set Δ is required. The values $L_\Delta(i,j)$ will appear explicitely in the constraints of the proposed mixed-integer formulations. The computation of the $L_\Delta(i,j)$ values can be done in polynomial time for some uncertainty sets, such as: budgeted uncertainty sets (Bertsimas and Sim 2004), uncertainty sets defined by a polynomial number of scenarii, and their convex hulls. It can also be done with a pseudo-polynomial algorithm for uncertainty sets with several uncertainty budget constraints introduced in (Minoux 2007). When the computation of the $L_\Delta(i,j)$ values can be done in a preprocessing step, then the proposed algorithmic framework can be applied using the $L_\Delta(i,j)$ values.

3 Two MIP formulations

3.1 A compact MIP formulation with anchoring and schedule variables

Theorem 1. *A valid formulation for (AnchRob) is*

$$
(\mathcal{F}_{zh}) \qquad
\begin{array}{lll}
\max \sum_{i\in J} a_i h_i & & \\
\text{s.t. } z_j - z_i \geq L_0(i,j) + (L_\Delta(i,j) - L_0(i,j))\, h_j & & \forall i \in \bar{J}, j \in \bar{J} \setminus \{t\} \\
\quad z_t - z_i \geq L_0(i,t) & & \forall i \in J \\
\quad z_t \leq M & & \\
\quad z_j \geq 0 & & \forall j \in \bar{J} \\
\quad h_j \in \{0,1\} & & \forall j \in J
\end{array}
$$

Note first that for a feasible pair (z,h) of $\mathcal{F}_{zh}$, the set H associated with h is anchored w.r.t. z, thanks to Proposition 1. The proof of the validity of formulation $\mathcal{F}_{zh}$ relies on a dominance argument: we prove that for any set H anchored w.r.t. some baseline schedule, there exists a baseline schedule z for which H is anchored and (z,h) is feasible for $\mathcal{F}_{zh}$. Note also that this formulation has a polynomial number of variables and constraints.

3.2 An exponential formulation with anchoring variables only

Let $\overline{G}$ be a weighted graph defined as the transitive closure of G, where every arc (i,j) has arc-length $L_\Delta(i,j)$ if $j \neq t$, and $L_0(i,t)$ otherwise. Let $\mathscr{P}$ (resp. $\mathscr{P}^{>M}$) denote the set of s–t paths of $\overline{G}$ (resp. the set of s–t paths of $\overline{G}$ that have length $> M$). Given $P \in \mathscr{P}$, let $V(P)$ denote the jobs of J along path P. We show the following result.

Theorem 2. *A valid formulation for (AnchRob) is*

$$
(\mathcal{F}_{h}) \qquad
\begin{array}{lll}
\max \sum_{i\in J} a_i h_i & & \\
\text{s.t. } \sum_{i\in V(P)} h_i \leq |V(P)| - 1 & \forall P \in \mathscr{P}^{>M} & \textit{(PathCov)} \\
\quad h_j \in \{0,1\} & \forall j \in J &
\end{array}
$$

The family of *path covering* inequalities (PathCov) imposes that there is at least one non-anchored job along any path of $\mathscr{P}^{>M}$. Note that it is necessary: if all jobs along the path were anchored, then with Proposition 1 every associated baseline schedule would have makespan $> M$. Formulation $\mathcal{F}_h$ has an exponential number of constraints, thus we study the separation of (PathCov) inequalities. The separation problem is a constrained longest path problem in $\overline{G}$, that is, the problem of finding a path with length $> M$, and sum of $1 - h_i$ over vertices at least 1. Formally, we show that

Theorem 3. *Separation of (PathCov) is weakly NP-hard and admits a pseudo-polynomial algorithm based on dynamic programming. Separation of (PathCov) is polynomial-time solvable in an integer point $h \in \{0,1\}^J$.*

We mention that, given $h \in \{0,1\}^J$ feasible for $\mathcal{F}_h$, a baseline schedule for which the corresponding set H is anchored can be found in polynomial time. Indeed it is sufficient to compute an earliest schedule that satisfies the precedence constraints from $G(p)$ and the precedence constraints from Proposition 1. In particular, this justifies that we obtained a valid formulation for (AnchRob) with only h variables.

4 Algorithmic framework

4.1 Comparison of formulations for budgeted uncertainty

Consider budgeted uncertainty. Then three formulations for (AnchRob) are available: formulations $\mathcal{F}_{zh}$ and $\mathcal{F}_h$, and the formulation from (Bendotti *et al.* 2019) denoted by $\mathcal{F}_{xh}$. It can be shown that the optimal values of the linear relaxation of $\mathcal{F}_{zh}$ and $\mathcal{F}_h$ are always better than the optimal value of the linear relaxation of $\mathcal{F}_{xh}$. We thus focused on $\mathcal{F}_{zh}$ and $\mathcal{F}_h$; it appears that the two formulations cannot be compared w.r.t. their linear relaxations.

Theorem 4. *$\mathcal{F}_h$ and $\mathcal{F}_{zh}$ are not comparable, i.e., there exists instances where the optimal value of the linear relaxation of $\mathcal{F}_h$ is better than the optimal value of the linear relaxation of $\mathcal{F}_{zh}$, and vice versa.*

We also provide an explicit formulation of the projection of formulations $\mathcal{F}_{zh}$ and $\mathcal{F}_{xh}$ on the space of anchoring variables. E.g., for $\mathcal{F}_{zh}$ formulation:

Proposition 2. *Let $h \in \{0,1\}^J$. There exists z such that (z,h) is feasible for $\mathcal{F}_{zh}$ if and only if h satisfies the inequalities $\sum_{(i,j)\in P} L_0(i,j) + (L_\Delta(i,j) - L_0(i,j))h_j \leq M \ \forall P \in \mathscr{P}$.*

4.2 Dedicated Branch-And-Cut algorithm

Theorem 4 suggests the use of a combination of $\mathcal{F}_h$ and $\mathcal{F}_{zh}$ to solve efficiently larger instances of (AnchRob). We formulate the problem with z and h variables. All constraints from $\mathcal{F}_{zh}$ are enforced in a static way. (PathCov) inequalities from $\mathcal{F}_h$ are separated in a heuristic way to strengthen the formulation. Namely, we only separate (PathCov) inequalities in integer points, thus in polynomial time.

Additional features are considered to improve the efficiency of separated (PathCov) inequalities, relying on polyhedral considerations. In particular, it can be shown that if inequality associated with path $P \in \mathscr{P}^{>M}$ is facet-defining, then it must satisfy: for any $i \in P$, the path $P' := P \setminus \{i\}$ is not an element of $\mathscr{P}^{>M}$. We enforce this property during the separation process: an inequality (PathCov) is separated, then vertices are removed from the path until the property is satisfied. We will give numerical results to illustrate the relevance of the proposed Branch-And-Cut algorithm.

References

P. Bendotti, Ph. Chrétienne, P. Fouilhoux, A. Quillot, 2017, "Anchored reactive and proactive solutions to the CPM-scheduling problem", *European Journal of Operational Research*, Vol. 261, pp. 67–74.

P. Bendotti, Ph. Chrétienne, P. Fouilhoux, A. Pass-Lanneau, 2019, "The Anchor-Robust Project Scheduling Problem", *Preprint*, `https://hal.archives-ouvertes.fr/hal-02144834`.

D. Bertsimas and M. Sim, 2004, "The Price of Robustness", *Operations Research*, Vol. 52, pp. 35–53.

M. Minoux, 2007, "Models and Algorithms for Robust PERT Scheduling with Time-Dependent Task Durations". *Vietnam Journal of Mathematics*, Vol. 35, pp 387–398.

M. Minoux, 2008, "Robust linear programming with right-handside uncertainty, duality and application", *Encyclopedia of Optimization*, Springer, pp. 3317–3327.