A two-stage robust approach for minimizing the weighted number of tardy jobs with profit uncertainty

Henri Lefebvre[1], François Clautiaux[2] and Boris Detienne[2]

[1] DEI, University of Bologna, Italy
henri.lefebvre@unibo.it
[2] IMB, University of Bordeaux, Inria Bordeaux Sud-Ouest, France
francois.clautiaux,boris.detienne@math.u-bordeaux.fr

Keywords: Single machine scheduling, Robust optimization, Exact algorithm.

1 Introduction

We investigate a stochastic variant of the well-know $1|r_j|\sum w_j U_j$ problem, in which the jobs are subject to unexpected failure which leads to additional costs. The decision maker is then allowed to take recourse actions such as outsourcing or spending more time on the jobs to fix them. We are interested in worst-case optimization, with polyhedral uncertainty set affecting the objective function.

In our problem, called *Two-Stage Robust Weighted Number of Tardy jobs* (2SRWNT) in the sequel, an instance consists of a set of jobs $\mathcal{J}$, each of which is characterized by a release date r_j, a due date d_j, and a nominal processing time p_j. A weight w_j can be interpreted as the cost for executing the job tardy, or the opposite of the profit of processing the job on time. At the first stage, *here-and-now* decisions are to select a subset of jobs $\mathcal{J}^* \subseteq \mathcal{J}$ to process. After that, a subset of the jobs can be affected by unexpected failures, those being governed by the uncertainty set $\varXi = \left\{ \xi \in \mathbb{R}_+^{|J|} \;\middle|\; \xi_j \le 1, \forall J_j \in \mathcal{J} \text{ and } \sum_{j|J_j\in\mathcal{J}} \xi_j \le \varGamma \right\}$. The realization of alea $\xi \in \varXi$ determines a profit degradation for each job $J_j \in \mathcal{J}$ defined as $\delta_j(\xi) = \bar{\delta}_j \xi_j$, where $\bar{\delta}_j$ is the maximum additional cost linked to the job's failure. Input parameter $\varGamma$ is the largest number of jobs that can incur their maximum degradation. At the second stage *recourse* actions have to be taken. For each $j \in \mathcal{J}^*$, one can choose *(i)* to keep the revealed profit ; *(ii)* to repair the job, adding τ_j time units to its processing time to recover its initial profit ; or *(iii)* to reject the job, and pay a fixed outsourcing cost f_j. Finally, jobs in $\mathcal{J}^*$ that are not rejected must be scheduled so that they meet their time windows. The objective is to select a subset of jobs as well as the recourse actions that minimize the worst-case overall cost (equivalently, maximizes the overall worst-case profit).

(van den Akker, Hoogeveen and Stoef 2018) also study a variant of $1||\sum U_j$ where the processing times are uncertain. Given a discrete scenario-based uncertainty set, one has to determine an initial, feasible for nominal processing times, sequence of jobs. At second stage, once the scenario of actual processing times is revealed, the sequence must be made feasible for those actual processing times by rejecting some jobs. The objective is to minimize the expected cost of the repaired solution. Exact methods are proposed for this problem. Our study differs by the basic problem (we consider unequal release dates and weights), the nature of the uncertainty set (polyhedral vs. discrete, scenario-based), the uncertain data (objective vs. constraints) and the possible recourse actions.

Robustness is known to be a hard issue in scheduling. (Aloulou and Della Croce 2008) and (Yang and Yu 2002) show that even simple scheduling problems become $\mathcal{NP}$-hard as soon as the uncertainty set contains more than one scenario. A possible way to address our problem is to use the so-called finite adaptability model of (Bertsimas and Caramanis 2010). This heuristic approach consists in restricting the problem by determining at first stage a set of K recourse solutions, while the second stage is reduced to choosing the best of those for the revealed alea. On the one hand, when K is small enough, this approach has the advantage to produce tractable problems. On the other hand, it may produce suboptimal solutions, since it restricts the number of resource actions that can be performed.

The contribution of this abstract is to propose the first exact method for this problem. It is based on an MILP formulation based on a recent result of (Arslan and Detienne 2018). We solve the model using a branch-and-price algorithm.

2 Mixed Integer Linear Programming model

We first recall the idea of the ILP model proposed in (Detienne 2014) for $1|r_j|\sum w_j U_j$, which we extend to the robust case. Their approach is based on the fact that minimizing the weighted number of tardy jobs can be decomposed into two distinct decisions: (1) decide which jobs are to be executed tardy and (2) in what order the on-time jobs are executed. We know that if the jobs have agreeable time windows (i.e., that the tasks can be ordered in such a way that for each J_i before J_j we have $r_i \leq r_j$ and $d_i \leq d_j$), then a feasible sequence of on-time jobs exists iff the earliest due-date first rule (EDD) yields a feasible solution. The main idea of (Detienne 2014) is to reformulate the general $1|r_j|\sum w_j U_j$ problem into a problem of selecting jobs with agreeable time windows. To do so, a set of so-called *job occurrences* is created from the original set of jobs in such a way that EDD may still be applied. Formally, consider a job $J_i \in \mathcal{J}$. For any job $J_j \in \mathcal{J}$ whose time window is not agreeable with that of J_i (i.e., $r_i < r_j$, $d_i > d_j$, and $r_i + p_i + p_j \leq d_j$), we create a job occurrence $J_k \in \widetilde{\mathcal{J}}$ such that $r_k = r_i, p_k = p_i, w_k = 0, f_k = f_i, \bar{\delta}_k = \bar{\delta}_j, \tau_k = \tau_j$ and a hard deadline $\bar{d}_k = d_j$ and which represents the scheduling of J_i before J_j. The original job J_i is also added to the set of job occurrences $\widetilde{\mathcal{J}}$, with a null weight as well. We define, for every job $J_j \in \mathcal{J}$, $\mathcal{G}_j$ as the set gathering all the job occurrences related to J_j. The following proposition, established in (Detienne 2014), naturally extends to the robust case.

Proposition 1. *There is at least one optimal solution of 2SRWNT such that selected job occurrences are scheduled according to a non-decreasing order of their deadlines with ties being broken in a non-decreasing order of their release dates.*

In the remainder, we assume that job occurrences are sorted according to a non-decreasing order of their deadlines and denote by $\bullet_k$ the data $\bullet$ of the kth occurrence in that order (e.g., p_k now denotes the processing time of the kth job occurrence in that order). Similarly to (Detienne 2014), we assign reversed time windows to each job occurrence given by $[\hat{r}_j, \hat{d}_j] = [\max_i d_i - d_j, \max_i d_i - r_j]$, which helps writing an ILP model with a stronger linear relaxation.

For every job $J_j \in \mathcal{J}$, we introduce decision variable U_j which is equal to 1 if J_j is tardy, 0 otherwise. For every job occurrence $J_k \in \mathcal{G}_j$, we denote by y_k the selection variable of the kth job occurrence, and z_k the decision variable indicating whether the job occurrence is repaired or not. More precisely, if $U_j = 0$, then J_j is decided to be executed on-time in the first stage. Once the uncertainty is revealed, the sequencing of the jobs and the recourse actions have to be decided. The following cases may arise: *(i)* $\exists J_k \in \mathcal{G}_j, y_k = z_k = 1$, *i.e.* the job is executed and repaired ; *(ii)* $\exists J_k \in \mathcal{G}_j, y_k = 1$ and $z_k = 0$, *i.e.* the job is executed and the deteriorated profit is undertaken ; *(iii)* $\forall J_k \in \mathcal{G}_j, y_k = z_k = 0$, the job is outsourced. Let us introduce the set $\mathcal{Y} \subset \{0,1\}^{|\widetilde{\mathcal{J}}|} \times \{0,1\}^{|\widetilde{\mathcal{J}}|} \times \mathcal{R}_+^{|\widetilde{\mathcal{J}}|}$ of all feasible second-stage solutions:

$$\mathcal{Y} = \begin{cases} \rho_k = p_k y_k + \tau_k z_k \quad \forall k | J_k \in \widetilde{\mathcal{J}} & (1) \\ z_k \leq y_k \quad \forall k | J_k \in \widetilde{\mathcal{J}} & (2) \\ \sum_{k | J_k \in \mathcal{G}_j} y_k \leq 1 \quad \forall J_j \in \mathcal{J} & (3) \\ \hat{t}_k + \rho_k - M_k(1 - y_k) \leq \hat{d}_k \quad \forall k | J_k \in \widetilde{\mathcal{J}} & (4) \\ \hat{t}_{k-1} - \hat{t}_k - \rho_k \geq 0 \quad \forall k \neq 1 | J_k \in \widetilde{\mathcal{J}} & (5) \\ \hat{t}_k \geq \hat{r}_k, \rho_k \geq 0 \quad \forall k | J_k \in \widetilde{\mathcal{J}} & (6) \\ y_k, z_k \in \{0,1\} \quad \forall k | J_k \in \widetilde{\mathcal{J}} & (7) \end{cases}$$

Here, $\hat{t}_k$ is the variable equal to the (reverse) starting time of occurrence k, while ρ_k is equal to the processing time of occurrence k, and M_k is a large constant. Constraints (1) define the processing time of a job with respect to the recourse action. Constraints (2) enforce that a job may be repaired only if it is scheduled. Constraints (3) limits the number of selected occurrences to one per job. Constraints (6) and (4) respectively enforce that no job starts before its release date or finish after its deadline, while constraint (5) makes sure that no two jobs overlap. By denoting $\mathcal{Y}(U) = \{(y, z, \hat{t}, \rho) \in \mathcal{Y} \mid \sum_{k|J_k \in \mathcal{G}_j} y_k \leq 1 - U_j \quad \forall J_j \in \mathcal{J}\}$ the set of feasible second-stage solutions that are consistent with first-stage solution U, the objective function is given by:

$$\min_{U \in \{0,1\}^{|\mathcal{J}|}} \sum_{j | J_j \in \mathcal{J}} w_j U_j + f_j(1 - U_j) + \max_{\xi \in \Xi} \min_{(y,z,\hat{t},\rho) \in \mathcal{Y}(U)} R(\xi, y, z)$$

where $R(\xi, y, z)$ denotes the cost of recourse action (y, z) corresponding to scenario ξ given by: $R(\xi, y, z) = \sum_{j|J_j \in \mathcal{J}} \sum_{k|J_k \in \mathcal{G}_j} \left[(\bar{\delta}_k \xi_j - f_k) y_k - \bar{\delta}_k \xi_j z_k\right]$. Note that the outsourcing cost has been moved to the first-stage: it is assumed that outsourcing is always paid for on-time jobs unless the job is scheduled in the second stage. Also, remark that for a given $U \in \{0,1\}^{|\mathcal{J}|}$ the recourse function $Q(U, \xi) = \min_{(y,z) \in \mathcal{Y}(U)} R(\xi, y, z)$ is not a convex function of ξ. That implies that the worst-case is in general not achieved at an extreme point of Ξ, so that more than Γ jobs might see their profit degraded by a small amount.

This formulation of 2SRWNT possesses interesting features. First, the uncertainty is polyhedral and only enters the objective function. Second, the constraints linking the first and second stages $\sum_{k|J_k \in \mathcal{G}_j} y_k \leq 1 - U_j \quad \forall J_j \in \mathcal{J}$ can be expressed as $\gamma \leq \beta$, with γ and β vectors of binary decision variables associated respectively with the second and first stage. This allows us to use the methodology introduced in (Arslan and Detienne 2018) to reformulate 2SRWNT, which is based on the following successive steps: (i) replacing $\mathcal{Y}(U)$ with its convex hull expressed in terms of its extreme points (using Minkowski-Weyl theorem) ; (ii) permuting the inner max and min (using von Neumann theorem) ; (iii) linearizing the inner max using LP duality (Bertsimas and Sim 2004). Denoting by $(\mathbf{y}^e, \mathbf{z}^e), e \in E$ the extreme points of conv $\mathcal{Y}$, we obtain the following MILP model:

$$(ColGen) : \min \sum_{J_j \in \mathcal{J}} [w_j U_j + f_j(1 - U_j) + v_j] + \Gamma u - \sum_{k | J_k \in \widetilde{\mathcal{J}}} \left[f_k \sum_{e \in E} \mathbf{y}_k^e \alpha_e \right]$$

$$\text{s.t.} \sum_{e \in E} \alpha_e = 1 \tag{8}$$

$$\sum_{e \in E} \mathbf{y}_k^e \alpha_e \leq 1 - U_j \quad \forall k | J_k \in \mathcal{G}_j, \forall J_j \in \mathcal{J} \tag{9}$$

$$u + v_j \geq \sum_{k | J_k \in \mathcal{G}_j} \left[\bar{\delta}_k \sum_{e \in E} (\mathbf{y}_k^e - \mathbf{z}_k^e) \, \alpha_e \right] \quad \forall j | J_j \in \mathcal{J} \tag{10}$$

$$U_j \in \{0,1\} \quad \forall J_j \in \mathcal{J}, \alpha_e \geq 0 \quad \forall e \in E, u \geq 0, v_j \geq 0 \quad \forall J_j \in \mathcal{J}$$

Here, decision vector α represents the convex combination multipliers from the reformulation of conv($\mathcal{Y}$) while u and v are the dual variables associated to the constraint $\xi \in \Xi$. Constraint (9) links the recourse action with the first-stage decision. Constraint (8) enforces that the recourse actions are convex combinations of the extreme points of conv($\mathcal{Y}$). Finally, constraint (10) embeds the dualized cost associated to the worst-case scenario.

Problem 2SRWNT is trivially $\mathcal{NP}$-hard. This formulation proves as a corollary, that quite surprisingly for a min-max-min problem with integer recourse, it lies inside class $\mathcal{NP}$ and is thus $\mathcal{NP}$-complete (Arslan and Detienne 2018).

3 Numerical experiments

We develop a branch-and-price algorithm to solve model (*ColGen*), based on the C++ library BapCod (Vanderbeck 2005). The pricing problem consists in finding a solution in $\mathcal{Y}$ minimizing the reduced-cost. This variant of $1|r_j|\sum w_j U_j$ with two possible modes per job (normal or repaired) is solved with a MILP solver. Our approach is compared against the finite adaptability method of (Hanasusanto, Kuhn and Wiesemann 2015), which is the method that is the closest to our ours, although it results in a heuristic formulation. We solve this model directly using a general purpose commercial solver.

We compare both approaches on a set of 3200 randomly generated instances. Our branch-and-price algorithm solves to optimality all 20 job-instances of our test bed within one hour, and 85% of the 25 job-instances. Our method provides as by-product, for each solved instance, the number K^* of recourse solutions required to achieve optimality. When using K^* as the parameter of the finite adaptability model, it fails at solving some 10 job-instances. It solves less than 17% of the instances for which $K^* \geq 2$ and $|I| = 25$.

4 Conclusion

We have proposed a numerically effective algorithm to solve a hard robust scheduling problem exactly. It compares favorably to the finite adaptability approach, in terms of computing time and quality of solutions.

References

Aloulou, M. A. and Della Croce, F.: 2008, Complexity of single machine scheduling problems under scenario-based uncertainty, *Oper. Res. Lett.* **36**(3), 338–342.

Arslan, A. N. and Detienne, B.: 2018, Reformulation approaches for a two-stage robust knapsack problem, *International workshop on robust optimization, 2018*, Avignon.

Bertsimas, D. and Caramanis, C.: 2010, Finite Adaptability in Multistage Linear Optimization, *IEEE Transactions on Automatic Control* **55**(12), 2751–2766.

Bertsimas, D. and Sim, M.: 2004, The Price of Robustness, *Operations Research* **52**(1), 35–53.

Detienne, B.: 2014, A mixed integer linear programming approach to minimize the number of late jobs with and without machine availability constraints, *European Journal of Operational Research* **235**, 540–552.

Hanasusanto, G. A., Kuhn, D. and Wiesemann, W.: 2015, K-adaptability in two-stage robust binary programming, *Operations Research* **63**(4), 877–891.

van den Akker, M., Hoogeveen, H. and Stoef, J.: 2018, Combining two-stage stochastic programming and recoverable robustness to minimize the number of late jobs in the case of uncertain processing times, *Journal of Scheduling* **21**(6), 607–617.

Vanderbeck, F.: 2005, Bapcod - a generic branch-and-price code.
URL: *https://realopt.bordeaux.inria.fr/?page_id=2*

Yang, J. and Yu, G.: 2002, On the robust single machine scheduling problem, *Journal of Combinatorial Optimization* **6**(1), 17–33.