An Almost Exact Multi-Machine Scheduling Solution for Homogeneous Processing

Stefano Nasini
IESEG School of Management, 3 rue de la Digue 59000 Lille, France.
LEM-CNRS 9221, 3 rue de la Digue, 59000 Lille, France.
s.nasini@ieseg.fr

Rabia Nessah
IESEG School of Management, 3 rue de la Digue 59000 Lille, France.
LEM-CNRS 9221, 3 rue de la Digue, 59000 Lille, France.
r.nessah@ieseg.fr

In the context of job scheduling in parallel machines, we present a class of asymptotically exact binary programs for the minimization of the τ-norm of completion time variances. Building on overlooked properties of the min completion time variance in a single machine and on an equivalent bilevel formulation, our approach provides an asymptotic approximation (with quadratic convergence) that can be computed much more efficiently than the original problem. Dominance properties are enforced as linear constraints to improve the characterization of the exact solution by these bounds. Our numerical study reveals that the lower bound problem can be solved in less than one third of computation time, in comparison to the exact problem, while resulting in an almost identical solution.

Key words: Multi-machine scheduling; Completion time variance; Bilevel optimization; Optimality bounds; Dominance properties.

1. Introduction

Driven by the interest in just-in-time production systems (Miltenburg 1989, Li et al. 2006) and project management (Van de Vonder et al. 2008, Balouka and Cohen 2019), the completion time lateness and variance have been used as non-regular performance measures (Oyetunji 2009) to target homogeneity in scheduling problems. This has given rise to important theoretical results in both single and multiple machines systems (Merten and Muller 1972, Eilon and Chowdhury 1977, Cai and Cheng 1998, Viswanathkumar and Srinivasan 2003, Li et al. 2010, Srirangacharyulu and Srinivasan 2010).

In the latter case, when n jobs have to be scheduled in m parallel machines, the min-sum and the min-max are standard aggregation rules (that we denote as $PM|\omega_i|\sum CTV$ and $PM|\omega_i|\max CTV$, using the standard notation by Graham et al. (1979)), laying in the two extreme points of the multi-criteria solution preferences (T'kindt and Billaut 2006). For the min-sum, Federgruen and Mosheiov (1996) proposed tight bounds to minimize the sum of completion time deviations from a common due date (a special case is the problem of minimizing the sum of completion time variances). For the min-max, Li and Cheng (1994) and Cheng et al. (1995) have first addressed the problem of minimizing the maximum absolute distance between completion times and a common due date.[1]

Integrating the min-sum and min-max rules, this paper studies a class of asymptotically exact binary programs for the minimization of the τ-norm of completion time variances of n jobs in m identical parallel machines. These bounds are formulated as binary quadratically constrained programs (hereafter referred to as BQCP), solvable by state-of-the-art branch-and-cuts algorithms.

While the vast majority of contributions on this stream of literature focus either on the characterization of dominance properties[2] or on the construction of efficient heuristics[3], a substantially smaller attention has been given to the formulation of either exact or asymptotically correct binary programs. This is probably due to the presence of symmetry conditions (driven by the identical machines) and weak LP relaxations (Beaumont 1997, Arora et al. 2003, Chlamtac and Tulsiani 2012), so that a massive amount of cuts is needed along the branch-and-bound tree to prune integer nodes. Recent openings toward polyhedral representations in different classes of scheduling problems are the works of Verschae and Wiese (2014), Correa et al. (2015) and Kurpisz et al. (2018), focusing on improving the linear integrality gaps.

Building on the asymptotic optimality in scheduling problems (Armony et al. 2019, Balseiro et al. 2018), we show that an equivalent bilevel program can translate the original problem of τ-norm minimization into BQCP computable bounds, exploiting overlooked properties of the min completion time variance in a single machine $1||CTV$. Specifically, we provide theoretical insights on three fundamental aspects of the proposed approach.

- Firstly, we prove the equivalence between a class of bilevel problems (a single-leader-multi-follower game), where machines act as followers subject to the centralized leader's decisions on job assignment, and their high-point relaxations $PM|\omega_i|\sum CTV$ and $PM|\omega_i|\max CTV$.[4]

- Secondly, this equivalent bilevel formulation allows exploiting overlooked properties of $1||CTV$, characterizing asymptotically exact bounds on the followers' best responses, by a system of quadratic constraints on binary variables. We show that the relative gap between these bounds converges quadratically to zero as the number of jobs increases.

- Thirdly, dominance properties are enforced as linear constraints to improve the characterization of the exact solution, as described by Jouglet and Carlier (2011). This has important consequences on the definition of symmetry-breaking constraints, to eliminate the presence of equivalent subproblems in the branch-and-bound tree.

On the empirical side, the modelling and resolution approach introduced in this paper has been designed to target specific scheduling applications, that are numerically studied in the final section of this paper. Hereafter, we introduce these applications.

- *Nursing home management.* Motivated by the growing interest in nursing management (Green et al. 2013), we consider the assignment of patients (elderly or disabled people) to the assistive personnel in residential care facilities. To build this data set, we interacted with a nursing home in Spain, accessing information about the daily assistance.

- *IT troubleshooting service.* Focusing on daily information from the IT troubleshooting service of a French educational institution, we consider the assignment of services (concerning software or hardware breakdown) to the IT personnel and their daily scheduling.

- *311 call centers.* We explore the assignment of personnel to calls in a non-emergency municipal services in the Chicago Data Portal, using online available data about the number of calls and available operators reaching a call center.

Using data about these three applications, our numerical study reveals that the lower bound problem can be solved in less than one third of computation time, in comparison to the exact problem, while resulting in an almost identical solution. This result suggests the use to the proposed lower bound as an efficient alternative to the exact minimization of the τ-norm of completion time variances in parallel machines

The rest of this paper is organized as follows. Section 2 sets the baseline model. Section 3 builds the lower and upper bounds and describes their asymptotic correctness. Section 3.1 proposes a collection of dominance properties of the original problems to be appended as linear constraints to the BQCP reformulations. A comprehensive computational test is provided in Section 4. Section 5 presents the conclusion of the paper. All mathematical proofs are reported in Appendix A.

2. Baseline model

In the context of job scheduling in parallel machines, a binary program for the minimization of the τ-norm of completion time variances is presented in this section. We introduce the following notation:

$\mathcal{M} = \{1, \ldots, m\}$:=a set of machines;
$\mathcal{J} = \{1, \ldots, n\}$:=a set of jobs to be performed;
$\mathcal{J}_i \subseteq \mathcal{J}$:=a set of jobs in the i^{th} machine (with $|\mathcal{J}_i| = n_i$ and $\sum_{i=1}^{m} n_i = n$);[5]
ω_i :=the initial availability moment of machine $i \in \mathcal{M}$;
p_h :=the processing time of job $h \in \mathcal{J}$.

Let σ_i be a sequence of n_i jobs and let $C_{h,i}(\sigma_i, \mathcal{J}_i)$ be the completion time of job h when the i^{th} machine is endowed with the jobs in $\mathcal{J}_i$ and are scheduled in accordance with the sequence σ_i. For a sequence σ_i and a collection of jobs $\mathcal{J}_i$, we can compute the mean completion time and the completion time variance (CTV from now on) as:

$$\overline{C}_i(\sigma_i, \mathcal{J}_i) = \frac{1}{n_i}\sum_{h=1}^{n_i} C_{h,i}(\sigma_i, \mathcal{J}_i) \qquad \text{and} \qquad CTV_i(\sigma_i, \mathcal{J}_i) = \frac{1}{n_i}\sum_{h=1}^{n_i}[C_{h,i}(\sigma_i, \mathcal{J}_i) - \overline{C}_i(\sigma_i, \mathcal{J}_i)]^2.$$

The problem of minimizing the τ-norm of completion time variances in parallel machines becomes

$$\theta^\tau(m) = \min_{(\sigma_i,\ \mathcal{J}_i)_{i\in\mathcal{M}}} \left(\sum_{i\in\mathcal{M}} CTV_i(\sigma_i, \mathcal{J}_i)^\tau\right)^{1/\tau}. \tag{1}$$

Note that this model covers an entire class of problems that has been separately studied.

Remark 1 (**Min-sum and min-max generalization**). Problem (1) generalises the min-sum and the min-max completion time variances of n jobs in m identical parallel machines. In fact, when $\tau = 1$ problem (1) reduces to $PM|\omega_i|\sum CTV$ and when $\tau = \infty$ problem (1) reduces to $PM|\omega_i|\max CTV$.

Next, we proceed to provide a BQCP formulation, as well as two BQCP computable bounds to (1). To do so, we introduce the binary indicator $y_{h,i,k}$, that is equal to one if job h is in the k^{th} position in machine i and zero otherwise. This binary variables must satisfy the following collection of equality constraints:

$$\begin{cases} \displaystyle\sum_{h=1}^{n} y_{h,i,k} = 1,\ \forall\, i\in\mathcal{M},\ k = 1\ldots n_i, \\ \displaystyle\sum_{i=1}^{m}\sum_{k=1}^{n_i} y_{h,i,k} = 1,\ \forall\, h\in\mathcal{J}. \end{cases} \tag{2}$$

Therefore, after the $\mathcal{J}_i$ decisions are taken, the processing times of jobs in machine $i\in\mathcal{M}$ becomes

$$x_{i,k} = \sum_{h=1}^{n} p_h y_{h,i,k}. \tag{3}$$

Similar binary and mixed-integer programming formulations for parallel machines scheduling have been proposed in the last two decades (Chang et al. 2004, Chung et al. 2009, Correa et al. 2015, Muter 2020). The advantage of this arrangement is that it can be used both to sort jobs in increasing order (as required for the generation of the lower and upper bounds next in this section) and to compute the completion times (as required for the generation of the exact solution). In both cases, using (2) and (3), the makespan[6] of each machine $i\in\mathcal{M}$ can be written as $MS_i(\mathbf{x}_i) = \omega_i + \sum_{k=1}^{n_i} x_{i,k}$ and its completion time variance can be written as

$$CTV_i(\mathbf{x}_i) = \frac{1}{n_i}\sum_{h=1}^{n_i}\left[\sum_{k=1}^{h}(\omega_i + x_{i,k}) - \frac{1}{n_i}\sum_{k=1}^{n_i}(\omega_i + x_{i,k})\right]^2 = \frac{1}{n_i}\sum_{h=1}^{n_i}\left[\sum_{k=1}^{h} x_{i,k} - \frac{1}{n_i}\sum_{k=1}^{n_i} x_{i,k}\right]^2,$$

where we replaced the notation $CTV_i(\sigma_i, \mathcal{J}_i)$ with $CTV_i(\mathbf{x}_i)$ to emphasize the dependency with respect to the decision variables of the BQCP. Problem (1) can be rewritten as

$$\theta^\tau(m) \; = \; \begin{cases} \min\limits_{\varphi, \mathbf{y}} \; \left(\sum_{i\in\mathcal{M}} \varphi_i\right)^{1/\tau} & \\ \text{s.t.} \;\; \varphi_i \;\geq\; [CTV(\mathbf{x}_i)]^\tau & \forall\, i \in \mathcal{M}, \\ \qquad \text{verifying (2) and (3)} & \forall\, i \in \mathcal{M}. \end{cases} \tag{4}$$

Consistently with Remark 1, the two specific cases of $\tau = 1$ (min-sum) and $\tau = \infty$ (min-max) result in the following programs:

$$\theta^1(m) = \begin{cases} \min\limits_{\mathbf{y}} \; \sum\limits_{i\in\mathcal{M}} CTV_i(\mathbf{x}_i) & \forall\, i \in \mathcal{M}, \\ \qquad \text{verifying (2) and (3)} \; \forall\, i \in \mathcal{M}, \end{cases} \tag{5}$$

and

$$\theta^\infty(m) = \begin{cases} \min\limits_{\varphi, \mathbf{y}} \; \varphi & \\ \text{s.t.} \;\; \varphi \geq CTV_i(\mathbf{x}_i) & \forall\, i \in \mathcal{M}, \\ \qquad \text{verifying (2) and (3)} \; \forall\, i \in \mathcal{M}, \end{cases} \tag{6}$$

For all the other values of τ, problem (4) can be approached by piecewise linear approximations, by defining a collection $\mathcal{T}$ of interpolation seeds with values ℓ_t, for $t \in \mathcal{T}$. This gives rise to the following BQCP computable problems:

$$\hat{\theta}^\tau(m) \; = \; \begin{cases} \min\limits_{\varphi, \mathbf{y}} \; \sum_{i\in\mathcal{M}} \varphi_i & \\ \text{s.t.} \;\; \varphi_i \;\geq\; (\ell_t)^\tau + \tau(\ell_t)^{\tau-1}\left[CTV_i(\mathbf{x}_i) - \ell_t\right] & \forall\, i \in \mathcal{M}, t \in \mathcal{T}, \\ \qquad \text{verifying (2) and (3)} & \forall\, i \in \mathcal{M}, \end{cases} \tag{7}$$

where we took advantage of the fact that $f(x) = x^{1/\tau}$ is strictly increasing. Therefore, $(\hat{\theta}^\tau_{UB}(m))^{1/\tau}$ and $\hat{\theta}^p_{LB}(m)$ respectively approach $\theta^\tau_{UB}(m)$ and $\theta^\tau_{LB}(m)$ when the the interpolation seeds increase.

3. Lower and upper bounds

Theoretical studies on the minimum CTV on a single machine (denoted as $1|\omega|CTV$, using the standard notation by Graham et al. (1979)) have provided dominance properties, as well as tight lower and upper bounds on the minimal variance.[7] To exploit these results for the minimization of the τ-norm of completion time variances in identical parallel machines, we establish an equivalence between (1) on the one hand, and a bilevel formulation on the other hand. The letter consists of a single-leader-multi-follower game capturing the sequential process of job assignments and scheduling by different players (a leader, in charge for assigning jobs to machines, and independent machines, in charge for the internal scheduling, subject to the first-stage leader decision). This equivalence allows translating well-known dominance properties of the minimum CTV on a single machine into a polyhedral representation of problem (1).

Proposition 1 **(Bilevel formulation)**. *Consider the following bilevel programming models:*

$$\hat{\theta}^{\tau}(m) \;=\; \min_{\mathcal{J}_1\cdots\mathcal{J}_m} \left(\sum_{i\in\mathcal{M}} CTV(\sigma_i^*(\mathcal{J}_i),\mathcal{J}_i)^{\tau} \right)^{1/\tau}, \tag{8}$$

where

$$\sigma_i^*(\mathcal{J}_i) \;=\; \underset{\sigma}{\operatorname{argmin}}\; CTV(\sigma,\mathcal{J}_i). \tag{9}$$

We claim that the optimal solution of (1) *is equivalent to the one of* (8).

The proof of Proposition 1 is in Appendix A.

Based on this analogy, we refer to the selection of job assignments $\mathcal{J}_1\ldots\mathcal{J}_m$ as *leader decision* and to the selection of job scheduling within each machine $\sigma_1\ldots\sigma_m$ as *followers' decision*. The next proposition establishes lower and an upper bounds for the best responses of each follower, subject to the centralized leader decisions on job assignments to machines.

Proposition 2. *For each follower $i\in\mathcal{M}$, let us enforce the increasing order of processing times by setting the following n_i-1 constraints for each machine $i\in\mathcal{M}$:*

$$x_{i,1}\le x_{i,2}\le\ldots\le x_{i,n_i}. \tag{10}$$

Once the $\mathcal{J}_1\ldots\mathcal{J}_m$ decisions are taken, the minimum CTV can be bounded as follows:

$$CTV_i(\sigma_i^*,\mathcal{J}_i)\le UB_i(\mathcal{J}_i)=\frac{1}{n_i}\left[\sum_{k=1}^{\alpha_i}\frac{1}{2}\left(\eta_{0,k}(\mathbf{x}_i)\right)^2+\sum_{k=\alpha_{0,i}}^{\alpha_i}2\left(\frac{1}{2}\eta_{1,k}(\mathbf{x}_i)-\eta_2(\mathbf{x}_i)\right)^2\right], \tag{11}$$

$$CTV_i(\sigma_i^*,\mathcal{J}_i)\ge LB_i(\mathcal{J}_i)=\frac{1}{2n_i}\left[\sum_{k=1}^{\alpha_i}\left(\eta_{0,k}(\mathbf{x}_i)\right)^2\right]. \tag{12}$$

- *If n_i is even, $\alpha_{0,i}=1$ and*

$$\eta_{0,k}(\mathbf{x}_i)=\sum_{j=1}^{2k-1}x_{i,j}, \tag{13}$$

$$\eta_{1,k}(\mathbf{x}_i)=2\omega_i+\sum_{j=0}^{\alpha_i-1}x_{i,n_i-2j}+\sum_{j=1}^{k}x_{i,2j-1}+\sum_{j=0}^{\alpha_i-k}x_{i,n_i-2j}, \tag{14}$$

$$\eta_2(\mathbf{x}_i)=\omega_i+\frac{1}{n_i}\sum_{k=1}^{\alpha_i}\sum_{j=0}^{k-1}x_{i,n_i-2j}+\frac{1}{n_i}\sum_{k=1}^{\alpha_i}\sum_{j=1}^{k}x_{i,2j-1}+\frac{1}{2}\sum_{j=0}^{\alpha_i-1}x_{i,n_i-2j}. \tag{15}$$

- *If n_i is odd, $\alpha_{0,i}=0$ and*

$$\eta_{0,k}(\mathbf{x}_i)=\sum_{j=1}^{2k}x_{i,j}, \tag{16}$$

$$\eta_{1,k}(\mathbf{x}_i) = 2\omega_i + \sum_{j=0}^{\alpha_i} x_{i,n_i-2j} + \sum_{j=1}^{k} x_{i,2j} + \sum_{j=0}^{\alpha_i-k} x_{i,n_i-2j}, \tag{17}$$

$$\eta_2(\mathbf{x}_i) = \omega_i + \frac{1}{n_i}\left(\sum_{k=1}^{\alpha_i+1}\sum_{j=k}^{\alpha_i+1} x_{i,2j} + (\alpha_i+1)\sum_{j=1}^{\alpha_i+1} x_{i,2j} + \sum_{k=1}^{\alpha_i+1}\sum_{j=1}^{k} x_{i,2j-1}\right). \tag{18}$$

Corollary 1. *For every $i \in \mathcal{M}$, when $n_i \le 4$ and $x_{i,1} \le x_{i,2} \le \ldots \le x_{i,n_i}$, we have*

$$CTV_i(\sigma_i^*, \mathcal{J}_i) = UB_i(\mathcal{J}_i)$$

and

$$UB_i(\mathcal{J}_i) - LB_i(\mathcal{J}_i) = \begin{cases} \frac{\left(x_{i,3}-x_{i,2}\right)^2}{16} & \text{if } n_i = 4 \\ \frac{\left(x_{i,2}-x_{i,1}\right)^2}{12} & \text{if } n_i = 3. \end{cases}$$

The proof of Proposition 2 and Corollary 1 are in Appendix A.[8]

Building on these bounds for the min completion time variance in a single machine $1|\omega_j|CTV$ and using the algebraic characterizations (2) and (3), we can invoke Proposition 1 to obtain the following BQCP computable bounds for (1). Therefore, piecewise linear approximated upper and lower bounds can be computed as follows:

$$\hat{\theta}^{\tau}_{UB}(m) = \begin{cases} \min\limits_{\varphi,\mathbf{y}} & \sum_{i\in\mathcal{M}} \varphi_i \\ \text{s.t.} & \varphi_i \;\ge\; (\ell_t)^{\tau} + \tau(\ell_t)^{\tau-1}\left[UB_i(\mathbf{x}_i) - \ell_t\right] \qquad \forall\, i \in \mathcal{M}, t \in \mathcal{T}, \\ & \text{verifying (2), (3), (10), (11), and (16)-(18)/(13)-(15)} \quad \forall\, i \in \mathcal{M}, \end{cases} \tag{19}$$

$$\hat{\theta}^{\tau}_{LB}(m) = \begin{cases} \min\limits_{\varphi,\mathbf{y}} & \sum_{i\in\mathcal{M}} \varphi_i \\ \text{s.t.} & \varphi_i \;\ge\; (\ell_t)^{\tau} + \tau(\ell_t)^{\tau-1}\left[LB_i(\mathbf{x}_i) - \ell_t\right] \qquad \forall\, i \in \mathcal{M}, t \in \mathcal{T}, \\ & \text{verifying (2), (3), (10), (12) and (16)/ (13)} \; \forall\, i \in \mathcal{M}, \end{cases} \tag{20}$$

where we replaced the notation $UB_i(\mathcal{J}_i)$ and $LB_i(\mathcal{J}_i)$ with $UB_i(\mathbf{x}_i)$ and $LB_i(\mathbf{x}_i)$ to emphasize the dependency with respect to the decision variables of the BQCP. For any τ, these formulations verify the relationship:

$$\hat{\theta}^{\tau}_{LB}(m) \;\le\; \hat{\theta}^{\tau}(m) \;\le\; \hat{\theta}^{\tau}_{UB}(m).$$

Next, a central aspect to be explored is the tightness of these bounds. To address it, we first consider the following property of the mean completion time of the followers' best responses.

Proposition 3. *Once the $\mathcal{J}_1 \ldots \mathcal{J}_m$ decisions are taken, for each machine $i \in \mathcal{M}$, let*

$$\sigma_i^*(\mathcal{J}_i) \;=\; \operatorname*{argmin}_{\sigma} \; CTV_i(\sigma, \mathcal{J}_i).$$

We claim that

$$L_i(\mathcal{J}_i) \;\le\; \overline{C}_i(\sigma_i^*(\mathcal{J}_i), \mathcal{J}_i) \;\le\; U_i(\mathcal{J}_i), \tag{21}$$

where

$$\begin{cases} L_i(\mathcal{J}_i) = \omega_i + \dfrac{1}{2}\left(\displaystyle\sum_{k=1}^{n_i} x_{i,k} + x_{n_i}\right) - \dfrac{n_i - 2}{2n_i}\left(x_{n_i-1} - x_{n_i-2}\right) \text{ and} \\ U_i(\mathcal{J}_i) = \omega_i + \dfrac{n_i - 1}{2n_i}\displaystyle\sum_{k=1}^{n_i} x_{i,k} + \dfrac{n_i+1}{2n_i}x_{n_i} - \dfrac{n_i-2}{4n_i}x_{n_i-1} + \dfrac{n_i+2}{4n_i}x_{n_i-2} + \dfrac{n_i-3}{2n_i}x_{n_i-3}. \end{cases} \tag{22}$$

The proof of Proposition 3 is in Appendix A.

Note that (21) induces a collection of valid inequalities for (1), which are not guaranteed to be verified by the solutions of (19) and (20). However, under certain regularity conditions, the relative gap between $U_i(\mathcal{J}_i)$ and $L_i(\mathcal{J}_i)$ goes to zero for all $i \in \mathcal{M}$.

PROPOSITION 4 **(Asymptotic correctness of the mean)**. *As long as $p_1 \dots p_n$ are uniformly bounded with respect to n, then for all $i \in \mathcal{M}$:*

$$\lim_{n_i \longrightarrow \infty} \frac{U_i(\mathbf{x}_i) - L_i(\mathbf{x}_i)}{U_i(\mathbf{x}_i)} \;\le\; \lim_{n_i \longrightarrow \infty} \left(\frac{(n_i+3)x_{i,n_i}}{\omega_i + n_i x_{i,1}}\right)\frac{1}{2n_i} \;=\; 0 \tag{23}$$

where we replaced the notation $U_i(\mathcal{J}_i)$ and $L_i(\mathcal{J}_i)$ with $U_i(\mathbf{x}_i)$ and $L_i(\mathbf{x}_i)$ to emphasize the dependency with respect to the decision variables of the BQCP.

The proof of Proposition 4 is not reported extensively, as it follows directly by simplifying the values in (22). In the same vein, using (21) and (22), the next proposition establishes the convergence of the relative gaps between $UB_i(\mathcal{J}_i)$ and $LB_i(\mathcal{J}_i)$.

PROPOSITION 5 **(Asymptotic correctness of the variance)**. *As long as $p_1 \dots p_n$ are uniformly bounded with respect to n, then for all $i \in \mathcal{M}$:*

$$\lim_{n_i \longrightarrow \infty} \frac{UB_i(\mathbf{x}_i) - LB_i(\mathbf{x}_i)}{UB_i(\mathbf{x}_i)} \;\le\; \lim_{n_i \longrightarrow \infty} \frac{\kappa_i}{x_{i,1}^2}\frac{2}{n_i^2} \;=\; 0 \tag{24}$$

where $\kappa_i = \max\{(x_{i,n_i} - \omega_i - x_{i,n_i-2})^2, (x_{i,n_i} + x_{i,n_i-3} - \omega_i)^2\}$ is a constant independent from n_i.

The proof of Proposition 5 is in Appendix A.

From Proposition 5, we know that problems (19) and (20) are asymptotically correct approximations of problem (1). Note that the validity condition of Propositions 4 and 5 is verified by the vast majority of applications, as the processing times of new jobs are independent from the number of already existing jobs.

Next to it, from Proposition 4 and Proposition 5, we see that the relative gaps of completion time means have linear convergence, while the ones of completion time variances have quadratic convergence. This implies that we expect a small relative gap even for medium size problems. (This will be numerically explored in Section 4.)

3.1. Linearly computable dominance properties

The construction of the upper and lower bounds (19) and (20) relies on the increasing order of processing times, enforced by constraints (10). Taking advantage of the same characterization, dominance properties are explored in this section to improve the proximity between the asymptotically exacts bounds (19) and (20) and the original problem (1). Indirectly, this has important consequences on the definition of symmetry-breaking constraints (to eliminate the presence of equivalent sub-problems in the branch-and-bound tree). In fact, given a feasible solution $\mathbf{y}$ for (19) and (20), a set of other $(m!-1)$ equivalent solutions can be generated by permuting $y_{h,i,k}$ with $y_{h,i',k}$ for all $h \in \mathcal{J}$ and $k = 1 \ldots n_i$, because all machines are identical.[9]

Hereafter a collection of symmetry-breaking constraints for problems (19) and (20) are deduced from dominant properties of problem (1). In the rest of this section, processing times are assumed to be sorted in increasing order $p_1 \leq p_2 \leq \ldots \leq p_n$. The following propositions describe simultaneous constraints that at least one optimal integer solution of problems (1) must verify, and which cannot be verified by the $(m!-1)$ replicates of the optimal solutions.

Proposition 6. *There exists at least one optimal solution of* (1) *verifying the following constraints:*

$$x_{i,n_i} = p_{n-i+1}, \qquad \text{for } i \in \mathcal{M}. \tag{25}$$

Proposition 7. *For each $i \in \mathcal{M}$, the optimal solution of problems* (1) *verifying* (25) *must also verify the following constraints:*

$$\sum_{k=1}^{n_i-1} x_{i,k} \leq \min\left\{ \sum_{k=1}^{n_i-1} p_{n-k+1},\ \frac{n_i-2}{2} x_{i,n_i-1} - \frac{n_i-6}{2} x_{i,n_i-2} + (n_i-3) x_{i,n_i-3} \right\}. \tag{26}$$

Proposition 8. *Let order the indexes of machines is such a way as $n_m < n_{m-1} < \ldots < n_1$. Any optimal solution of problem* (1) *verifying* (25) *must also verify the following constraint:*

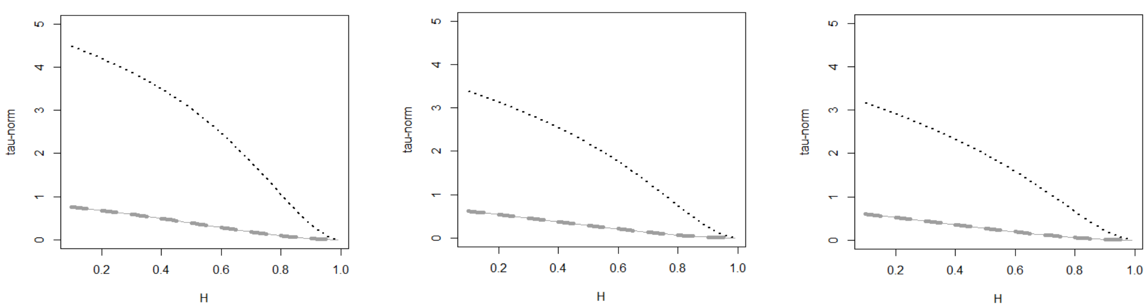
$$\sum_{i=1}^{m} x_{i,n_i} + \sum_{i=1}^{m} x_{i,n_i-1} \geq \sum_{i=1}^{m} p_{n-i+1} + \sum_{i=1}^{m} p_{n-m-\sum_{s=1}^{i-1} n_i - 1}. \tag{27}$$

The simultaneous linear constraints (25), (26) and (27) are valid for the original problem (1) and allow removing a large amount of sub-problems (which become infeasible) from the branch-and-bound tree. However, constraints (25) and (26) are not guaranteed to be verified by the optimal solutions of (19) and (20), so that their inclusion into these formulations might enforce a different job-machine assignment, improving the bounds, as numerically assessed in the next section.

3.2. Illustrative example

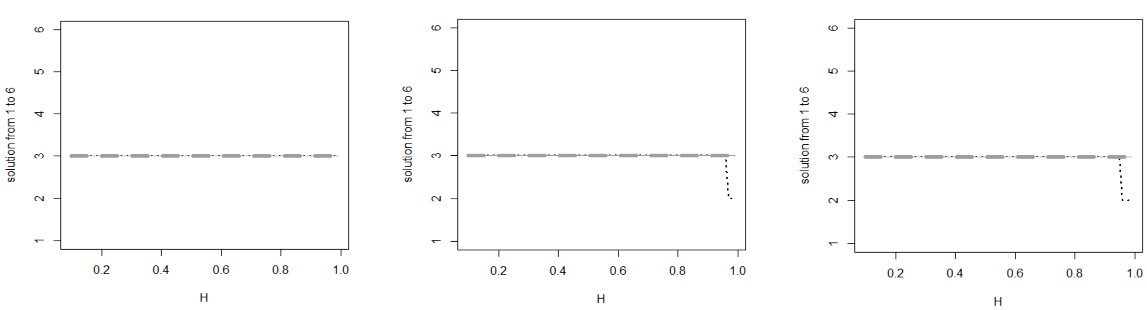
To visualize the relationship between the lower and upper bound solutions, as well as the combinatorial structure of the problem, we consider a job processing system with $m = 2$, $n_1 = 3$, $n_2 = 3$, and define the processing times $p_1 \leq p_2 \leq \ldots \leq p_6$. Based on Corollary 1, the upper bound and exact objective function (from problem (4)) coincide and the six feasible assignment-scheduling solutions can be enumerated. For the first machine, we have: $(6,1,2),(6,1,3),(6,1,4),(6,2,3),(6,2,4),(6,3,4)$.

To provide a graphical illustration, we let $p_j = 1 - H^{j-1}$ and plot the differences between the exact solution and the lower bound in Figure 1. Contextually, Figure 2 reports the corresponding optimal solution from one to six, in the same order.



(a) $\tau = 1$. (b) $\tau = 2$. (c) $\tau = 3$.

Figure 1 **For different values of $H \in [0,1]$, the black dotted line and the gray dashed line denote the upper and lower bounds of the minimal τ-norm of CTV, respectively.**



(a) $\tau = 1$. (b) $\tau = 2$. (c) $\tau = 3$.

Figure 2 **For different values of $H \in [0,1]$, the black dotted line and the gray dashed line denote the optimal solution (from one to six in the respective order $(6,1,2),(6,1,3),(6,1,4),(6,2,3),(6,2,4),(6,3,4)$) of the upper and lower bound problems.**

Consistently, with Corollary 1, this toy example reveals the consistency of the lower and upper bound approximations for problems with $n_i \leq 4$. A similar figure is expected for the large limit of n_i, by means of Proposition 4 and Proposition 5, as numerically assessed in the next section.

4. Computational and empirical analysis

This section provides an empirical numerical support to the theoretical framework developed thus far. This is done by constructing a large scale computational experiment with several configurations of jobs and machines, based on three applications presented in Section 1. These candidate applications are described in details in the next subsection, while the two subsequent subsections are structured as follows.

- In Subsection 4.1, we explore the solvability of the upper and lower bound problems (19) and (20), as well as the exact problem (7) in the three empirical contexts of *nursing home management*, *IT troubleshooting service* and *311 call center*. Instances are simulated to mirror the data structures described in the next subsection.

- In Subsection 4.2, we explore the solvability of the upper and lower bound problems (19) and (20), as well as the exact problem (7) in simulated large-scale instances. We consider a collection of 9 instances with 5, 10, 20, 40, 60, 100, 150, 200, and 300 jobs per machine in two machines, to assess the asymptotic correctness of the upper and lower bound problems (19) and (20) and the opportunity of their use when the exact computation become challenging.

Both tests are performed with and without using the linear constraints (25), (26) and (27) and setting two different exit mipgap levels: $1e-4$ and $1e-1$. As far as the processing time generator is concerned, we consider a uniform distribution between b_i^- and b_i^+, for the i^{th} job category. In fact, in each of the three introduced applications, processing times can only be inferred from a given category of a given job (such as the dependency level of a patient or the type of request of a user).

All optimization procedures are solved using IBM ILOG CPLEX 12:9 implementation of the Branch-and-cut algorithm on a R5500 work-station with processor Intel(R) Xeon(R) CPU E5645 2.40 GHz, and 48 Gbytes of RAM, under a Windows Server 2012 operative system.

4.1. Numerical tests in real assignment-scheduling contexts

The solvability of the upper and lower bound problems (19) and (20), as well as the exact problem (7) is explored hereafter in the three empirical contexts of *nursing home management*, *IT troubleshooting service* and *311 call center*. Each instance is solved with and without appending constraints (25), (26) and (27). For the case of the exact formulation (7), the only constraints that have been appended is (25), as the dominance properties for (26) and (27) rely on the ordering (10), as established in Proposition 2 to characterized the bounds.

Nursing home management We collected information from a private residential care facility in Barcelona (Spain), hosting 72 elderly patients, who are daily supported by 8 nursing assistants. Each assistant is assigned to a group of patients, who must be sequentially assisted from eight o'clock in the morning. The processing times depend on the degree of dependency of each patient and have been estimated by Laures-Euskadi (2015), using data from 1.433 patients from six nursing homes in the Basque Country (Spain). Table 1 summarizes the aggregate information concerning the level of dependency and the approximate processing time at the residential care facility in Barcelona.

	Dependency 1	Dependency 2	Dependency 3
Number of patients	26	22	24
Declared processing time	7	12	20

Table 1 Partition of the 72 patients into dependency groups and time (in minutes) for each dependency level.

The processing time of each dependency level has been inferred by combining the publicly available data from Laures-Euskadi (2015) with the specific feedback obtained by interviewing the 8 nursing assistants in the residential care facility in Barcelona. This type of processing time measurement is not part of the working protocol in nursing homes. Instead, the dependency level is a common piece of information, which is used for financial purposes when it come to get access to public foundlings.

The generation of instances for this application mirrors the data structures reported in Table 1. For the i^{th} dependence category, the processing times of patients in that category are uniformly distributed between $b_i(1-\sigma)$ and $b_i(1-\sigma)$, where b_i is the reported processing time for the i^{th} dependency level of a patient and σ is an uncertainty parameter that we replicated at multiple values in the numerical tests.

Table 2 reports the CPU time and objective function of 16 instances. To compare the solution values provided by the lower problem (20), the upper bound problem (19) and the exact problem (7), Tables 3 reports the values of the min and the max among the smallest jobs: $\min\{\min\{x_{i,j} : j = 1\dots n_i\} : i \in \mathcal{M}\}$ and $\max\{\min\{x_{i,j} : j = 1\dots n_i\} : i \in \mathcal{M}\}$, respectively.

					without (25), (26), and (27)				appending (25), (26) and (27)			
			Exact (7)		Lower (20)		Upper (19)		Lower (20)		Upper (19)	
mipgap	τ	T	CPU	OF	CPU	OF	CPU	OF	CPU	OF	CPU	OF
0.100	1	1	12.5	5712.7	2.8	5670.0	6.0	5871.7	5.3	5682.9	7.0	5677.7
0.100	2	4	28.3	1414.5	5.5	1413.6	11.2	1415.7	6.2	1416.5	11.2	1417.6
0.100	2	6	30.4	1416.2	5.2	1413.7	11.1	1425.5	6.2	1420.2	12.4	1415.3
0.001	1	1	3661.1	5662.7	4.1	5660.1	3659.4	5663.1	6.8	5659.1	5358.9	5663.3
0.001	2	4	3650.7	1413.6	6.5	1413.1	3830.1	1413.6	7.5	1413.2	7220.9	1413.7
0.001	2	6	3727.6	1413.6	6.8	1413.1	3657.1	1413.6	10.9	1413.2	4972.2	1413.7

Table 2 CPU time and objective function of 16 instances solved five times each. Each row has been averaged over two instances, corresponding to two different levels of σ. The first three columns reports the information about the specification of the mipgap, τ and T. The subsequent columns are partitions in three groups. From left to right, each group contains the CPU time and objective function for the exact solution, the lower bound and the upper bound, with and without appending (25), (26) and (27).

				without (25), (26), and (27)				appending (25), (26) and (27)			
		Exact (7)		Lower (20)		Upper (19)		Lower (20)		Upper (19)	
mipgap	τT	max	min	max	min	max	min	max	min	max	min
0.100	11	6.8	5.2	6.8	5.2	7.2	5.2	7.2	5.2	7.2	5.2
0.100	24	7.0	5.2	6.6	5.2	7.0	5.2	7.1	5.2	7.1	5.2
0.100	26	7.1	5.2	6.7	5.2	7.2	5.2	7.1	5.2	7.2	5.2
0.001	11	6.6	5.2	7.1	5.2	7.0	5.2	6.6	5.2	7.1	5.2
0.001	24	6.6	5.2	6.6	5.2	6.8	5.2	6.6	5.2	6.8	5.2
0.001	26	6.7	5.2	6.6	5.2	6.8	5.2	6.7	5.2	7.1	5.2

Table 3 **The min and the max among the smallest jobs:** $\min\{\min\{x_{i,j}\ :\ j=1\dots n_i\}\ :\ i\in\mathcal{M}\}$ **and** $\max\{\min\{x_{i,j}\ :\ j=1\dots n_i\}\ :\ i\in\mathcal{M}\}$**, respectively. Each row has been averaged over two instances, corresponding to two different levels of** σ**. The first three columns reports the information about the specification of the mipgap,** τ **and** T**. The subsequent columns are partitions in three groups. From left to right, each group contains the min and the max among the smallest jobs for the exact solution, the lower bound and the upper bound, with and without appending (25), (26) and (27).**

The core insight from Table 2 is that the resolution of the lower bound problem (20) can be carried out with less than one third of computation time, in comparison to the exact problem (7). Next, a clear result from Table 3 is that the inclusion of the dominance properties (25), (26) and (27) has an impact on the concordance between the solution of the lower problem (20), the upper bound problem (19) and the exact problem (7), suggesting that these properties are violated by the solution of the bound reformulations. Then, solving (20) provides a solution that is almost entirely analogous (and asymptotically identical, based on propositions 4 and 5) to the one obtained by solving the exact problem (7). The advantage is that this solution is obtained with a mush smaller computational effort.

IT troubleshooting service Focusing on daily information from an IT troubleshooting service in a French academic institution, we consider the assignment of service requests (concerning software or hardware breakdown) to the IT personnel and their daily scheduling. These requests are sent through an internal software and received by a centralized office of the IT department. Table 4 contains average information about the IT troubleshooting tasks carried out in 2019.

	First semester	Second semester
Number of job categories	13	13
Average number of workers	22	24
Average number of jobs per day	10.46	11.75

Table 4 **Daily requests and available operators at the IT troubleshooting service.**

Generally, among the entire group of operators no more than two are simultaneously available to receive a service requests in a given day, as the others operators are frequently involved in general maintenance activities that are not related to individual service requests sent through the internal software. Therefore, these 10/12 daily requests must be assigned to the 2 available workers.

The generation of instances for this application mirrors the data structures reported in Table 4. We consider $n = 10$ and $n = 12$ daily service requests for the first and second semester respectively. We set $m = 2$ available operators. Since this instances are quite small, we can solve them to optimality by setting the mipgam to zero (to force the B&B algorithm to branch till the optimal node) and $T = 1000$ (to design a granular representation of τ-norm). The processing times have been simulated as for the previous case, with uniform distribution between $b_i(1-\sigma)$ and $b_i(1-\sigma)$, where b_i is the reported processing time for the i^{th} category and σ is an uncertainty parameter.

As for the previous case, tables 8 and 9 respectively report the performance and the solution properties of the lower problem (20), the upper bound problem (19) and the exact problem (7).

				without (25), (26), and (27)				appending (25), (26) and (27)			
		Exact (7)		Lower (20)		Upper (19)		Lower (20)		Upper (19)	
Semester	τ	CPU	OF	CPU	OF	CPU	OF	CPU	OF	CPU	OF
First	1	0.2	70.7	0.1	70.6	0.2	70.7	0.1	70.6	0.2	70.7
First	2	20.3	48.3	6.4	48.3	20.1	48.4	7.6	48.3	18.6	48.4
First	3	20.6	42.0	9.0	42.0	30.2	42.1	7.5	42.0	31.0	42.1
Second	1	0.2	103.3	0.1	103.3	0.2	103.3	0.1	103.3	0.1	103.3
Second	2	30.9	67.6	7.7	67.6	32.6	67.6	12.9	67.6	39.8	67.6
Second	3	30.9	57.2	9.2	57.2	34.7	57.2	11.5	57.2	31.7	57.2

Table 5 CPU time and objective function of 12 instances solved five times each. Each row has been averaged over two instances, corresponding to two different levels of σ. The first two columns reports the information about the specification of the data (corresponding to $n = 10$ and $n = 12$), and τ. The subsequent columns are partitions in three groups. From left to right, each group contains the CPU time and objective function for the exact solution, the lower bound and the upper bound, with and without appending (25), (26) and (27).

				without (25), (26), and (27)				appending (25), (26) and (27)			
		Exact (7)		Lower (20)		Upper (19)		Lower (20)		Upper (19)	
Semester	τ	max	min	max	min	max	min	max	min	max	min
First	1	3.6	3.3	3.5	3.3	3.6	3.3	3.5	3.3	3.6	3.3
First	2	3.6	3.3	3.5	3.3	3.6	3.3	3.5	3.3	3.6	3.3
First	3	3.6	3.3	3.5	3.3	3.6	3.3	3.5	3.3	3.6	3.3
Second	1	3.5	3.3	3.5	3.3	3.5	3.3	3.5	3.3	3.5	3.3
Second	2	3.5	3.3	3.5	3.3	3.5	3.3	3.5	3.3	3.5	3.3
Second	3	3.5	3.3	3.5	3.3	3.5	3.3	3.5	3.3	3.5	3.3

Table 6 The min and the max among the smallest jobs: $\min\{\min\{x_{i,j} \ : \ j = 1 \ldots n_i\} \ : \ i \in \mathcal{M}\}$ and $\max\{\min\{x_{i,j} \ : \ j = 1 \ldots n_i\} \ : \ i \in \mathcal{M}\}$, respectively. Each row has been averaged over two instances. The first two columns concern the data specification (corresponding to $n = 10$ and $n = 12$), and τ. The subsequent columns are partitions in three groups. From left to right, each group contains the min and the max among the smallest jobs for the exact solution, the lower bound and the upper bound, with and without appending (25), (26) and (27).

Also for this application to the IT troubleshooting service, we observe the same solution pattern. Specifically, solving (20) provides a solution that is almost identical to the one obtained by solving the exact problem (7), with a mush smaller computational effort.

311 call centers The *311* phone number provides access to non-emergency municipal services in the United State and Canada. This type of call centers face high congestion, so that multiple call can arrive within a tiny interval of time. The assignment and scheduling of these almost simultaneous calls (for instance, calls arriving in the same minute) can be done based on the τ-norm minimization problem described in in this article, with a view to achieve homogeneity in their completion times. To do so, we consider data from the Chicago Data Portal (2017) about the number of calls and the number of operators reaching a call center, as reported in Table 7.

	7:00AM-3:00PM	3:00PM-11:00PM	11:00PM-7:00AM
Average call volume	31678.6	22781.3	4545.7
Average call per minute	66.0	47.5	9.5
Average number of operators	20.0	9.4	2.6

Table 7 Call center data set, reporting the average call volume, the average call per minute and the average number of available operators at three different times of the day.

The generation of instances for this application mirrors the data structures reported in Table 7. Also in this case we set the mipgap to zero. Table 8 and 9 are structured as for the previous case, to respectively report the performance and the solution properties of the lower problem (20), the upper bound problem (19) and the exact problem (7).

					without (25), (26), and (27)				appending (25), (26) and (27)			
			Exact (7)		Lower (20)		Upper (19)		Lower (20)		Upper (19)	
Time	τ	T	CPU	OF	CPU	OF	CPU	OF	CPU	OF	CPU	OF
7:00AM-3:00PM	1	1	7200.0	– –	2.3	2144.7	3.5	2235.9	4.8	2253.0	3.6	2237.9
7:00AM-3:00PM	2	4	7200.0	– –	4.5	474.9	40.0	488.8	15.6	498.4	1667.9	500.6
7:00AM-3:00PM	2	6	7200.0	– –	2.5	486.1	28.1	493.2	28.1	496.7	105.3	503.5
3:00PM-11:00PM	1	1	7.9	1323.2	0.9	1262.3	0.9	1354.6	0.8	1320.8	1.7	1360.3
3:00PM-11:00PM	2	4	8.4	401.8	1.7	388.8	2.3	391.5	2.6	399.3	2.8	411.9
3:00PM-11:00PM	2	6	18.8	403.8	1.2	388.9	3.6	390.6	2.7	406.3	3.9	409.4
11:00PM-7:00AM	1	1	0.1	39.6	0.0	38.4	0.1	38.5	0.0	38.3	0.0	38.5
11:00PM-7:00AM	2	4	0.1	22.7	0.0	22.5	0.1	22.4	0.0	22.4	0.0	22.5
11:00PM-7:00AM	2	6	0.1	22.5	0.0	22.5	0.1	22.2	0.0	22.2	0.1	22.5

Table 8 CPU time and objective function of 18 instances solved five times each. Each row has been averaged over two instances, corresponding to two different levels of σ. The first three columns reports the information about the specification of the time, τ and T. The subsequent columns are partitions in three groups. From left to right, each group contains the CPU time and objective function for the exact solution, the lower bound and the upper bound, with and without appending (25), (26) and (27).

				without (25), (26), and (27)				appending (25), (26) and (27)			
		Exact (7)		Lower (20)		Upper (19)		Lower (20)		Upper (19)	
Time	τT	max	min	max	min	max	min	max	min	max	min
7:00AM-3:00PM	11	– –	– –	13.9	4.2	15.5	4.2	15.9	4.2	15.5	4.2
7:00AM-3:00PM	24	– –	– –	13.6	4.2	14.2	4.2	14.3	4.2	13.6	4.2
7:00AM-3:00PM	26	– –	– –	13.9	4.2	14.2	4.2	14.3	4.2	14.8	4.2
3:00PM-11:00PM	11	8.5	3.7	8.8	3.7	8.4	3.7	9.5	3.7	9.1	3.7
3:00PM-11:00PM	24	8.8	3.7	8.4	3.7	8.4	3.7	9.0	3.7	10.3	3.7
3:00PM-11:00PM	26	10.0	3.7	8.4	3.7	8.8	3.7	9.0	3.7	9.6	3.7
11:00PM-7:00AM	11	4.6	3.7	4.6	3.7	4.7	3.7	4.4	3.7	4.6	3.7
11:00PM-7:00AM	24	4.6	3.7	4.7	3.7	4.7	3.7	4.6	3.7	4.7	3.7
11:00PM-7:00AM	26	4.4	3.7	4.7	3.7	4.4	3.7	4.5	3.7	4.5	3.7

Table 9 The min and the max among the smallest jobs: $\min\{\min\{x_{i,j} : j=1\ldots n_i\} : i\in\mathcal{M}\}$ **and** $\max\{\min\{x_{i,j} : j=1\ldots n_i\} : i\in\mathcal{M}\}$**, respectively. Each row has been averaged over two instances, corresponding to two different levels of** σ**. The first three columns reports the information about the specification of the time,** τ **and** T**. The subsequent columns are partitions in three groups. From left to right, each group contains the min and the max among the smallest jobs for the exact solution, the lower bound and the upper bound, with and without appending (25), (26) and (27).**

In this case, with an integrality tolerance of $1.0e-5$, CPLEX cannot find a feasible integer solution of the exact problem (7) within the time limit of 7200 seconds. For the rest, analogous figures as for the previous applications remain valid: the lower problem (20) appears again as the most promising alternative to approach an *almost exact* solution of the τ-norm minimization problem.

4.2. Solving large-scale instances using the lower bound problem

Focusing on three real applications, the previous subsection showed that the resolution of the problems (19) and (20) can be carried out much more efficiently than the exact problem (7), with little (if any) differences in the obtained solution.

Building on this insight, this subsection explores the possibility of using problems (19) and (20) to deal with large-scale instances. Specifically, we consider the cases of $\tau=1$ (i.e., resulting in the min-sum problem $PM|\omega_i|\sum CTV$ in Remark 1) and $\tau=\infty$ (i.e., resulting in the min-max problem $PM|\omega_i|\sum CTV$ and $PM|\omega_i|\max CTV$ in Remark 1). For these two cases the piece-wise linearizations used in (7), (19), (20) are no longer required, so that the number of constraints drop substantially.

Tables 10 and 11 report the performance and solution of a collection of 16 instances corresponding to different values of n and τ.

		without (25), (26), and (27)				appending (25), (26) and (27)			
		Lower (20)		Upper (19)		Lower (20)		Upper (19)	
n	τ	CPU	OF	CPU	OF	CPU	OF	CPU	OF
90	1	6.0	308000.9	9.6	308177.1	6.7	307963.4	12.6	308250.8
90	∞	7.5	102734.7	21.1	102716.2	10.5	102715.9	18.1	102754.3
180	1	84.6	4786603.1	112.3	4787137.6	97.0	4786599.8	110.2	4788153.6
180	∞	101.1	1596483.1	104.1	1596453.0	106.3	1596435.6	113.5	1596527.7
270	1	235.7	23948940.8	302.1	23954220.3	302.1	23948968.0	308.2	23954637.9
270	∞	228.3	7989383.5	302.6	7987900.4	238.9	7985861.3	285.8	7986631.5
360	1	741.5	75157501.2	827.6	75164880.7	1831.8	75157762.2	2103.1	75168443.8
360	∞	755.2	25067991.1	849.1	25070888.1	1264.9	25064469.9	1627.9	25065129.2

Table 10 CPU time and objective function of 16 instances solved four times each. Each row has been averaged over two instances, corresponding to two different levels of σ. The first three columns reports the information about the specification of the time, τ and T. The subsequent columns are partitions in three groups. From left to right, each group contains the CPU time and objective function for the exact solution, the lower bound and the upper bound, with and without appending (25), (26) and (27).

		without (25), (26), and (27)				appending (25), (26) and (27)			
		Lower (20)		Upper (19)		Lower (20)		Upper (19)	
n	τ	max	min	max	min	max	min	max	min
90	1	3.3	1.2	5.3	1.2	3.3	1.2	4.3	1.2
90	∞	3.3	1.2	3.8	1.2	3.3	1.2	3.8	1.2
180	1	3.2	1.2	3.8	1.2	3.2	1.2	3.8	1.2
180	∞	4.8	1.2	6.4	1.2	4.8	1.2	5.3	1.2
270	1	3.2	1.2	7.4	1.2	3.2	1.2	7.4	1.2
270	∞	3.2	1.2	4.8	1.2	4.3	1.2	4.3	1.2
360	1	3.2	1.2	5.9	1.2	3.2	1.2	4.8	1.2
360	∞	3.2	1.2	4.8	1.2	3.2	1.2	4.8.4	1.2

Table 11 The min and the max among the smallest jobs: $\min\{\min\{x_{i,j} \;:\; j=1\dots n_i\} \;:\; i\in\mathcal{M}\}$ and $\max\{\min\{x_{i,j} \;:\; j=1\dots n_i\} \;:\; i\in\mathcal{M}\}$, respectively. Each row has been averaged over two instances, corresponding to two different levels of σ. The first three columns reports the information about the specification of the time, τ and T. The subsequent columns are partitions in three groups. From left to right, each group contains the min and the max among the smallest jobs for the exact solution, the lower bound and the upper bound, with and without appending (25), (26) and (27).

As for the three application in Subsection 4.1, also in the case of larger instances the figure is the one of an efficient lower level problem and a substantial benefit from constraints (25), (26) and (27) in the characterization of the solution.

5. Conclusions

In the context of job scheduling in identical parallel machines, this work exploited the equivalence between a class of bilevel scheduling problems (where machines act as followers, subject to the centralized leader decisions on job assignments), and their high-point relaxations. This allowed taking advantage from overlooked properties of $1||CTV$, characterizing asymptotically exact bounds on the followers' best responses, by a system of quadratic constraints on binary variables. This characterizations are valid for the problem of minimizing the τ-norm of completion time variances in m parallel machines, that represent a generalization of the min-sum and min-max objectives.

We proved the linear and the quadratic convergence of the respective completion time means and the completion time variances, providing a theoretical support to the goodness of the approximation of the proposed bounds to the exact problem.

Contextually, dominance properties are deduced and enforced as linear constraints to improve the characterization of the underlying solution.

Two numerical tests have been conducted, using a battery of 18 instances of $PM|\omega_i|\sum CTV$ and $PM|\omega_i|\max CTV$ and replicated over four different reformulations and resolution methods (the lower and upper bounds, with and without symmetry-breaking constraints). These tests reveal two main insights.

- Firstly, constraints (25), (26) and (27) are violated by the solutions of (19) and (20), in a large number of instances. This implies that regardless of the tightness of bounds for $1||CTV$, part of the missing information relates to the job assignment in the multi-machine configuration. At the same time, the inclusion of these dominance properties led to a decrease in the computational effort.

- Secondly, the resolution of the lower bound problem (20)can be solved in less than one third of computation time, in comparison to the exact problem, while resulting in an almost identical solution (as theoretically supported by Proposition 5).

Overall, our results show that the methodology we propose in this paper works extremely well for two very challenging combinatorial problems, which have been rarely solved up to optimality for large instances. Contextually, this methodology opens new possibilities for solving large instances of min-max and min-sum scheduling problems, beyond the CTV cases studied in this paper. A challenging area in which these developments may have a strong impact concerns stochastic counterpart of $PM|\omega_i|\sum CTV$ and $PM|\omega_i|\max CTV$, where multi-scenario reformulations can be considered. On this respect, decomposition algorithms (Codato and Fischetti 2006) can be studied and combined with specialized interior-point methods (Castro et al. 2017, Castro and Nasini 2017), to exploit the separability of scenario-dependent variables, once the combinatorial decisions are taken.

Appendix A: Mathematical Proof

Auxiliary lemmas

Before reporting the collection of mathematical proofs of the propositions presented in the manuscript, we consider the following working lemmas, which will be invoke at different points.

Without loss of generality, for every sequence σ, define as $\mathcal{P}(\sigma)$ the order set of processing times of jobs in σ.

Definition 1 (Follower Alternating Schedule $\sigma_{n_i}^{AS}$). For any leader decision $\mathcal{J}_1 \dots \mathcal{J}_m$, we define the *follower alternating schedule* for each machine $i \in \mathcal{M}$, as a sequence in which we schedule the largest job x_{i,n_i} to the first position, the second largest job x_{i,n_i-1} to the last position, the third largest job x_{i,n_i-2} to the second position, and so on as so forth alternatively untill $x_{i,1}$.

Lemma 1. *Let us consider a positive-real-function f defined on $\mathcal{X}$ ($f: \mathcal{X} \to \mathbb{R}_+$) and let x^* be the minimum of f over $\mathcal{X}$. Then x^* is also a minimum of the function $g(x) = (f(x)^p + c)^{1/p}$, for any arbitrary constant c.*

Proof of Lemma 1. If x^* is not a minimum of $g(x) = (f(x)^\tau + c)^{1/\tau}$, then there must exist $y^* \in \mathcal{X}$ such that

$$g(x^*) \;=\; (f(x^*)^\tau + c)^{1/\tau} \;>\; (f(y^*)^\tau + c)^{1/\tau} \;=\; g(y^*).$$

This inequality implies that $f(x^*) > f(y^*)$, which is in contradiction with respect to the optimality of x^*. Thus, x^* is a minimum of g over $\mathcal{X}$.

■

Lemma 2 (**Hall and Kubiak (1991)**). *Once the $\mathcal{J}_1 \dots \mathcal{J}_m$ decisions are taken, for each machine $i \in \mathcal{M}$, let*

$$\sigma_i^*(\mathcal{J}_i) \;=\; \operatorname*{argmin}_{\sigma} CTV(\sigma, \mathcal{J}_i).$$

We claim that

$$\mathcal{P}(\sigma_i^*(\mathcal{J}_i)) = (x_{i,n_i},\; x_{i,n_i-2}, \dots,\; x_{i,n_i-1})$$

In other words, the largest job is scheduled in the first position, the second largest in the last position and the third largest in the second position.

Lemma 3 (**Schrage (1975)**). *For a given machine $i \in \mathcal{M}$ and for a fixed assignment of processing times $\mathcal{J}_i$, consider the sequence $\sigma_i = (j_1, j_2, j_3, \dots, j_{r-1}, j_r)$. The dual sequence of σ_i is defined as $\sigma_i^d = (j_1, j_r, j_{r-1}, \dots, j_3, j_2)$. In other words, the first element of the sequence is kept in the same position, while the remaining d elements are inverted from top to bottom. We have the relationships*

$$\overline{C}_i(\sigma_i) + \overline{C}_i(\sigma_i^d) = MS_i(\mathbf{x}_i) + \omega_i + x_{i,n_i} \qquad \textit{and} \qquad CTV_i(\sigma_i) = CTV_i(\sigma_i^d). \tag{28}$$

Main propositions

Proof of Proposition 1. To show that the optimal solution of (1) is equivalent to the one of (8), we need to prove that

$$\gamma^J \;=\; \min_{(\sigma_i)_{i\in\mathcal{M}}} \left(\sum_{i\in\mathcal{M}} CTD(\sigma_i,\mathcal{J}_i,d_i)^p\right)^{1/p} \;=\; \left(\sum_{i\in\mathcal{M}} \min_{\sigma_i} CTV(\sigma_i,\mathcal{J}_i)^p\right)^{1/p},$$

for any fixed decision on job assignment $\mathcal{J}=(\mathcal{J}_1,...,\mathcal{J}_m)$. To do so, fix any machine $i\in\mathcal{M}$ and denote by $-i$ all machines except i. Let σ_i^* be the minimum of $CTV(\sigma_i,\mathcal{J}_i)$ and

$$\gamma^J(\sigma_{-i}) = \min_{\sigma_i} \left(CTV(\sigma_i,\mathcal{J}_i)^p + c(\sigma_{-i})\right)^{1/p},$$

where $c(\sigma_{-i})$ does not depend on σ_i. By Lemma 1, we deduce that

$$\gamma^J(\sigma_{-i}) = \left(CTV(\sigma_i^*,\mathcal{J}_i)^p + c(\sigma_{-i})\right)^{1/p},$$

By induction (and with the same process) on machine set $-i$, we obtain that

$$\gamma^J \;=\; \min_{(\sigma_i)_{i\in\mathcal{M}}} \left(\sum_{i\in\mathcal{M}} CTD(\sigma_i,\mathcal{J}_i,d_i)^p\right)^{1/p} \;=\; \left(\sum_{i\in\mathcal{M}} \min_{\sigma_i} CTV(\sigma_i,\mathcal{J}_i)^p\right)^{1/p},$$

which implies that the optimal solution of (1) is equivalent to the one of (8).

■

Proof of Proposition 2. Since the proof is valid for any machine $i\in\mathcal{M}$, we fix the arbitrary machine i and let $\alpha_i = \lfloor \frac{n_i}{2} \rfloor$ and $C_{(h)}(\sigma_i,\mathcal{J}_i)$ be the completion time of job in position h of machine i in the schedule σ. We equivalently use the short notation $C_{(k)}$, $\overline{C}_i$ and CTV_i to refer $C_{(k)}(\sigma,\mathcal{J}_i)$, $\overline{C}_i(\sigma_i,\mathcal{J}_i)$ and $CTV_i(\sigma_i,\mathcal{J}_i)$, respectively. Then, we have the following decomposition of the completion time variance:

- If n_i is even,

$$\begin{aligned}
CTV_i &= \frac{1}{n_i}\sum_{k=1}^{n_i}\left(C_{(k)}-\overline{C}_i\right)^2 \\
&= \frac{1}{n_i}\left(\sum_{k=1}^{\alpha_i}\left(C_{(k)}-\overline{C}_i\right)^2 + \sum_{k=\alpha_i+1}^{n_i}\left(C_{(k)}-\overline{C}_i\right)^2\right) \\
&= \frac{1}{n_i}\sum_{k=1}^{\alpha_i}\left(\left(C_{(\alpha_i+k)}-\overline{C}_i\right)^2 + \left(C_{(\alpha_i+1-k)}-\overline{C}_i\right)^2\right) \\
&= \frac{1}{n_i}\sum_{k=1}^{\alpha_i}\left(\frac{\left(C_{(\alpha_i+k)}-C_{(\alpha_i+1-k)}\right)^2}{2} + 2\left(\frac{1}{2}\left(C_{(\alpha_i+k)}+C_{(\alpha_i+1-k)}\right)-\overline{C}\right)^2\right).
\end{aligned}$$

- If n_i is odd

$$
\begin{aligned}
CTV_i &= \frac{1}{n_i}\sum_{k=1}^{n_i}\left(C_{(k)}-\overline{C}_i\right)^2 \\
&= \frac{1}{n_i}\left(\sum_{k=1}^{\alpha_i}\left(C_{(k)}-\overline{C}_i\right)^2+\sum_{k=\alpha_i+1}^{n_i}\left(C_{(k)}-\overline{C}_i\right)^2\right) \\
&= \frac{1}{n_i}\left[\left(C_{(\alpha_i+1)}-\overline{C}_i\right)^2+\sum_{k=1}^{\alpha_i}\left(\left(C_{(\alpha_i+k+1)}-\overline{C}_i\right)^2+\left(C_{(\alpha_i+1-k)}-\overline{C}_i\right)^2\right)\right] \\
&= \frac{1}{n_i}\left[\sum_{k=1}^{\alpha_i}\left(\frac{\left(C_{(\alpha_i+1+k)}-C_{(\alpha_i+1-k)}\right)^2}{2}+\sum_{k=0}^{\alpha_i}2\left(\frac{1}{2}\left(C_{(\alpha_i+1+k)}+C_{(\alpha_i+1-k)}\right)-\overline{C}_i\right)^2\right)\right].
\end{aligned}
$$

Note that in the alternating schedule σ^{AS} the smallest job $x_{i,1}$ is scheduled either at position $\alpha+1$, when n_i is odd, or at position $\alpha+2$, when n_i is even. Thus, for the case of the alternating schedule, each of the above terms of the CTV can be expressed as partial summations of the processing times as follows:

- If n_i is even,

$$
\begin{aligned}
\eta_{0,k}(\mathbf{x}_i) &= C_{(\alpha_i+k)}-C_{(\alpha_i+1-k)} &&= \sum_{j=1}^{2k-1}x_{i,j} \\
\eta_{1,k}(\mathbf{x}_i) &= C_{(\alpha_i+k)}+C_{(\alpha_i+1-k)} &&= 2\omega_i+\sum_{j=0}^{\alpha_i-1}x_{i,n_i-2j}+\sum_{j=1}^{k}x_{i,2j-1}+\sum_{j=0}^{\alpha_i-k}x_{i,n_i-2j} \\
\eta_2(\mathbf{x}_i) &= \overline{C}_i &&= \omega_i+\frac{1}{n_i}\sum_{k=1}^{\alpha_i}\sum_{j=0}^{k-1}x_{i,n_i-2j}+\frac{1}{n_i}\sum_{k=1}^{\alpha_i}\sum_{j=1}^{k}x_{i,2j-1}+\frac{1}{2}\sum_{j=0}^{\alpha_i-1}x_{i,n_i-2j}.
\end{aligned}
$$

- If n_i is odd

$$
\begin{aligned}
\eta_{0,k}(\mathbf{x}_i) &= C_{(\alpha_i+1+k)}-C_{(\alpha_i+1-k)} &&= \sum_{j=1}^{2k}x_{i,j} \\
\eta_{1,k}(\mathbf{x}_i) &= C_{(\alpha_i+1+k)}+C_{(\alpha_i+1-k)} &&= 2\omega_i+\sum_{j=0}^{\alpha_i}x_{i,n_i-2j}+\sum_{j=1}^{k}x_{i,2j}+\sum_{j=0}^{\alpha_i-k}x_{i,n_i-2j} \\
\eta_2(\mathbf{x}_i) &= \overline{C}_i &&= \omega_i+\frac{1}{n_i}\left(\sum_{k=1}^{\alpha_i}\sum_{j=0}^{\alpha_i-k}x_{i,n_i-2j}+(\alpha_i+1)\sum_{j=0}^{\alpha_i}x_{i,2j+1}+\sum_{k=1}^{\alpha_i}\sum_{j=1}^{k}x_{i,2j}\right).
\end{aligned}
$$

As a consequence, we deduce

$$
CTV_i(\sigma^*,\mathcal{J}_i)\le CTV_i(\sigma^{AS},\mathcal{J}_i)=\frac{1}{n_i}\left[\sum_{k=1}^{\alpha_i}\frac{1}{2}(\eta_{0,k}(\mathbf{x}_i))^2+\sum_{k=\alpha_{0,i}}^{\alpha_i}2\left(\frac{1}{2}\eta_{1,k}(\mathbf{x}_i)-\eta_2(\mathbf{x}_i)\right)^2\right],
$$

where $\alpha_{0,i}=0$, when n_i is even, and $\alpha_{0,i}=1$, when n_i is odd. To prove (12), it is sufficient to note that for any sequence σ

$$C_{(\alpha_i+k)}(\sigma,\mathcal{J}_i)-C_{(\alpha_i+1-k)}(\sigma,\mathcal{J}_i)\geq\sum_{j=1}^{2k-1}x_{i,j},$$
$$C_{(\alpha_i+k+1)}(\sigma,\mathcal{J}_i)-C_{(\alpha_i+1-k)}(\sigma,\mathcal{J}_i)\geq\sum_{j=1}^{2k}x_{i,j},$$

for the cases of n_i being even and odd respectively. Therefore, since $\left(\frac{1}{2}\eta_{1,k}(\mathbf{x}_i)-\eta_{0,k}(\mathbf{x}_i)\right)^2\geq 0$, we obtain

$$CTV(\sigma^*,\mathcal{J}_i)\geq\frac{1}{n_i}\sum_{k=1}^{\alpha_i}\eta_{0,k}(\mathbf{x}_i).$$

■

Proof of Corollary 1. Using the $1||CTV$ characterization from Hall and Kubiak (1991) (see Lemma 2 in Appendix A), we know that the optimal solution is in the form

$$\mathcal{P}(\sigma_i^*(\mathcal{J}_i))=\begin{cases}(x_{i,3},\ x_{i,1},\ x_{i,2}) & \text{if } n_i=3\\(x_{i,4},\ x_{i,2},\ x_{i,1},\ x_{i,3}) & \text{if } n_i=4\end{cases}$$

By construction, $UB_i(\mathcal{J}_i)$ (in Proposition 2) is the variance of the alternating scheduling. Therefore, $CTV_i(\sigma_i^*,\mathcal{J}_i)=UB_i(\mathcal{J}_i)$.

■

Proof of Proposition 3. Since the proof is valid for any machine $i\in\mathcal{M}$, we fix an arbitrary machine i and, for any sub-sequence $Q_i\subseteq\sigma_i$, we define $C_{ji}(Q_i)$, $\overline{C}_i(Q_i)$ and $CTV_i(Q_i)$, as the completion time of the j-th job in Q_i, the completion times mean of jobs in Q_i and the completion times variance of jobs in Q_i, respectively. Note that every time a sequence σ_i is decomposed in two sub-sequences π_i (corresponding to jobs $i_1\ldots i_k$) and S_i (corresponding to jobs $j_1\ldots j_s$), we can write

$$\overline{C}_i(\sigma_i)\ =\ \frac{k}{n_i}\overline{C}(\pi_i)+\frac{n_i-k}{n_i}\overline{C}_i(S_i)\quad\text{and}\quad\sum_{h=1}^{n_i}C_{h,i}^2(\sigma_i)\ =\ \sum_{h=1}^{k}C_{i_h,i}^2(\pi_i)+\sum_{h=1}^{n_i-k}C_{j_h,i}^2(S_i)$$

so that

$$\begin{aligned}
CTV_i(\sigma_i) &= \frac{1}{n_i}\sum_{h=1}^{n_i} C_h^2(\sigma_i) - \overline{C}^2(\sigma_i) \\
&= \frac{k}{n_i}\left[\sum_{h=1}^{k}\frac{C^2_{i_h,i}(\pi_i)}{k}\right] + \frac{n_i-k}{n_i}\left[\sum_{h=1}^{n_i-k}\frac{C^2_{j_h,i}(S)}{n_i-k}\right] - \left(\frac{k}{n_i}\overline{C}_i(\pi_i) + \frac{n_i-k}{n_i}\overline{C}_i(S_i)\right)^2 \\
&= \frac{k}{n_i}\left[CTV_i(\pi_i) + \overline{C}_i^2(\pi_i)\right] + \frac{n_i-k}{n_i}\left[CTV_i(S_i) + \overline{C}_i^2(S_i)\right] - \left(\frac{k}{n_i}\overline{C}_i(\pi_i) + \frac{n_i-k}{n_i}\overline{C}_i(S_i)\right)^2 \\
&= \frac{k}{n_i}CTV_i(\pi_i) + \frac{n_i-k}{n_i}CTV_i(S_i) + \frac{k(n_i-k)}{n_i^2}\left(\overline{C}_i(S_i) - \overline{C}_i(\pi_i)\right)^2.
\end{aligned}$$

Since we are focusing on a given machine, to lighten the notation, the index i and the dependency of C, $\overline{C}$ and CTV on the leader decisions will be dropped in the rest of this proof. The rest of the proof is considering the optimal sequence from Lemma 2:

$$\sigma^* = \Bigg(\underbrace{n_i}_{\pi_1},\ \underbrace{(n_i-2,\ j_1,\ j_2,\ \ldots,\ j_{n_i-3})}_{S},\underbrace{n_i-1}_{\pi_2}\Bigg). \tag{29}$$

Using the above variance decomposition and the notion of optimal sequence (30), the rest of this proof will first consider $L(\mathbf{x}_i)$ and then $U(\mathbf{x}_i)$.

Proof of $L(\mathbf{x}_i)$ Let us consider the sequence:

$$\sigma_i' = \Bigg(\underbrace{n_i}_{\pi_{i,1}},\ \underbrace{(n_i-2,\ j_{n_i-3},\ j_{n_i-4},\ \ldots,\ j_1)}_{S_{i,d}},\underbrace{n_i-1}_{\pi_{i,2}}\Bigg) \tag{30}$$

where $S_{i,d}$ is the dual sequence of S_i in Lemma 2. Building on Lemma 3 (by properly redefining the MS_i and the $x_{j_1,i}$), we obtain an analogous result for our sub-sequences $S_{i,d}$ and S in (29), from the previously described variance decomposition:

$$\begin{aligned}
\overline{C}_i(S_i) + \overline{C}_i(S_{i,d}) &= (MS_i - x_{n_i-1,i}) + (\omega_i + x_{n_i} + x_{n_i-2,i}), \\
CTV_i(\sigma^*) &\le CTV_i(\sigma') \\
&= \tfrac{2}{n_i}CTV_i(\pi_i) + \tfrac{n_i-2}{n_i}CTV_i(S_{i,d}) + \tfrac{2(n_i-2)}{n_i^2}\left(\overline{C}(S_{i,d}) - \overline{C}_i(\pi_i)\right)^2.
\end{aligned}$$

Using (28) and the fact that $\overline{C}(\pi) = (MS_i + \omega_i + x_{n_i})/2$, we can write $\overline{C}(S) + \overline{C}(S_d) = 2\overline{C}(\pi) - \Delta$, where $\Delta = x_{n_i-1} - x_{n_i-2}$, so that

$$\begin{aligned}
\left(\overline{C}(S) - \overline{C}(\pi)\right)^2 &\le \left(\overline{C}(S_d) - \overline{C}(\pi)\right)^2 \\
\left(\overline{C}(S) - \tfrac{\Delta + \overline{C}(S) + \overline{C}(S_d)}{2}\right)^2 &\le \left(\overline{C}(S_d) - \tfrac{\Delta + \overline{C}(S) + \overline{C}(S_d)}{2}\right)^2 \\
\left(\overline{C}(S) - \overline{C}(S_d) - \Delta\right)^2 &\le \left(\overline{C}(S_d) - \overline{C}(S) - \Delta\right)^2 \\
-2\Delta\left(\overline{C}(S) - \overline{C}(S_d)\right) &\le -2\Delta\left(\overline{C}(S_d) - \overline{C}(S)\right) \\
\overline{C}(S_d) &\le \overline{C}(S)
\end{aligned}$$

Therefore, $MS - x_{n_i-1} + \omega_i + x_{n_i} + x_{n_n-2} = \overline{C}(S) + \overline{C}(S_d) \leq 2\overline{C}(S)$ and hence

$$\begin{aligned}\overline{C}(\sigma^*) &= \tfrac{2}{n_i}\overline{C}(\pi) + \tfrac{n_i-2}{n_i}\overline{C}(S)\\ &\geq \tfrac{1}{n_i}\left(MS + \omega_i + x_{n_i} + (n_i-2)\tfrac{MS - x_{n_i-1} + \omega_i + x_{n_i} + x_{n_i-2}}{2}\right)\\ &\geq \tfrac{1}{2}\left(MS + \omega_i + x_{n_i}\right) - \tfrac{n_i-2}{2n_i}\left(x_{n_i-1} - x_{n_i-2}\right).\end{aligned}$$

Therefore,

$$\overline{C}(\sigma^*) \geq \frac{1}{2}\left(MS + \omega_i + x_{n_i}\right) - \frac{n_i-2}{2n_i}\left(x_{n_i-1} - x_{n_i-2}\right).$$

Proof of $U(\mathbf{x}_i)$ Let us consider the sequence:

$$\sigma'' = \left(n_i,\ n_i - 2, \underbrace{(j_2,\ \ldots,\ j_{n_i-3})}_{S''},\ j_1, n_i - 1\right) \tag{31}$$

which has been obtained from the optimal sequence σ^*, by switching job i_1 with the block S'' (keeping all the the jobs in S in their original relative positions). Denote by $z = x_{j_1}$ and compute the difference between the variances: By definition, we have

$$CTV(\sigma'') - CTV(\sigma^*) = \frac{1}{n_i}\sum_{h=1}^{n_i} C_h^2(\sigma'') - \overline{C}^2(\sigma'') - \frac{1}{n_i}\sum_{h=1}^{n_i} C_h^2(\sigma^*) + \overline{C}^2(\sigma^*). \tag{32}$$

We have the following relations between the two scheduling σ^* and σ'':

$$C_{j_h}(\sigma'') = \begin{cases} C_{j_h}(\sigma^*) & \text{if } h = n_i, n_i-1, n_i-2\\ C_{j_h}(\sigma^*) - z, & \text{if } h = 2, \ldots, n_i-3\\ MS - x_{n_i-1}, & \text{if } h = 1\end{cases}$$

which implies

$$\overline{C}(\sigma'') = \overline{C}(\sigma^*) + \frac{\gamma - (n_i-3)z}{n_i},$$

where $\gamma = MS - \omega - x_{n_i} - x_{n_i-1} - x_{n_i-2}$. Based on (32), we compute the following differences:

$$\begin{aligned}\textstyle\sum_{h=1}^{n_i}\left(C_h^2(\sigma'') - C_h^2(\sigma^*)\right) &= -2n_i z\overline{C}(\sigma^*) + (n_i-4)z^2 + 2z(\gamma + 4\omega + 4x_{n_i} + x_{n_i-1} + 3x_{n_i-2} + z)\\ &\quad + (\gamma - x)(\gamma + 2\omega + 2x_{n_i} + 2x_{n_i-2} + z)\end{aligned}$$

$$\overline{C}^2(\sigma'') - \overline{C}^2(\sigma^*) = 2\frac{\gamma - (n_i-3)z}{n_i}\overline{C}(\sigma^*) + \left(\frac{\gamma - (n_i-3)z}{n_i}\right)^2.$$

Therefore, we have

$$CTV(\sigma'') - CTV(\sigma^*) = -2\frac{\gamma + 3z}{n_i}\overline{C}(\sigma^*) + \frac{1}{n_i}f(z) \geq 0$$

where

$$f(z) = 3\underbrace{\frac{n_i-3}{n_i}}_{a} z^2 + 2\underbrace{(3\omega + 3x_{n_i} + x_{n_i-1} + 2x_{n_i-2} + \frac{2n_i-3}{n_i}\gamma)}_{b} z + \gamma\underbrace{(2\omega + 2x_{n_i} + 2x_{n_i-2} + \frac{n_i-1}{n_i}\gamma)}_{c}$$

which (by Euclidean division between $f(z)$ and $(3z+\gamma)$) implies

$$\overline{C}(\sigma^*) \leq g(z) = \frac{f(z)}{2(3z+\gamma)} = \frac{1}{2}\left(\frac{3az^2+2bz+\gamma c}{3z+\gamma}\right) = \frac{1}{2}\left(az + \frac{2b-a\gamma}{3} + \gamma\frac{c-\frac{2b-a\gamma}{3}}{3z+\gamma}\right). \tag{33}$$

Since g is increasing and z is bounded from above by x_{n_i-3}, then $g(z) \leq g(x_{n_i-3})$. Since $\frac{\gamma}{x_{n_i-3}} \geq 1$, then we get

$$\overline{C}(\sigma^*) \;\leq\; \frac{n_i-1}{2n_i}MS + \frac{n_i+1}{2n_i}(x_{n_i}+\omega) - \frac{n_i-2}{4n_i}x_{n_i-1} + \frac{n_i+2}{4n_i}x_{n_i-2} + \frac{n_i-3}{2n_i}x_{n_i-3}.$$

■

Proof of Proposition 5. Since we are focusing on a given machine, to lighten the notation, the index i and the dependency of C, $\overline{C}$ and CTV on the leader decisions will be dropped in the rest of this proof, when its usage is not needed.

Firstly, consider the CTV of the alternating schedule as a function of the completion time mean:

$$CTV(\sigma^{AS}, \overline{C}(\sigma^{AS})) = \underbrace{\frac{1}{n_i}\sum_{k=1}^{\alpha_i}(\eta_{0,k}(\mathbf{x}))^2}_{B_0} + \underbrace{\frac{1}{n_i}\sum_{k=\alpha_{0,i}}^{\alpha_i} 2\left(\frac{1}{2}\eta_{1,k}(\mathbf{x}) - \overline{C}(\sigma^{AS})\right)^2}_{B_1(\overline{C}(\sigma^{AS}))} \tag{34}$$

and note that to prove (24) it is sufficient to prove the following limit:

$$\lim_{n_i\longrightarrow\infty} \frac{\sum_{k=\alpha_{0,i}}^{\alpha_i}\left(\frac{1}{2}\eta_{1,k}(\mathbf{x}) - \overline{C}(\sigma^{AS})\right)^2}{\sum_{k=1}^{\alpha_i}(\eta_{0,k}(\mathbf{x}))^2} \;=\; \lim_{n_i\longrightarrow\infty} \frac{B_1(\overline{C}(\sigma^{AS}))}{B_0} \;=\; 0$$

Since $CTV(\sigma^{AS}, \overline{C}(\sigma^{AS}))$ is the average square deviation around $\overline{C}(\sigma^{AS})$, we know that for any real value d, we have

$$CTV(\sigma^{AS}, \overline{C}(\sigma^{AS})) \;\leq\; CTV(\sigma^{AS}, d)$$

which implies

$$\frac{B_1(\overline{C}(\sigma^{AS}))}{B_0} \;\leq\; \frac{B_1(\overline{C}(\sigma^*))}{B_0}.$$

From Proposition 3, when n_m grows large, we know that

$$\begin{cases} \overline{C}_i(\sigma_i^*) \;\geq\; \omega_i + \frac{1}{2}(MS_i + x_{n_i-2}) \\ \overline{C}_i(\sigma_i^*) \;\leq\; \omega_i + \frac{1}{2}MS_i + x_{n_i} + x_{n_i-3} \end{cases} \tag{35}$$

and from Proposition 2 we know that

$$\eta_{1,k}(\mathbf{x}) = \begin{cases} C_{(\alpha_i+k)} + C_{(\alpha_i+1-k)} = 2\omega_i + \sum_{j=0}^{\alpha_i-1} x_{n_i-2j} + \sum_{j=1}^{k} x_{2j-1} + \sum_{j=0}^{\alpha_i-k} x_{n_i-2j} & \text{when } n_i \text{ is even,} \\ C_{(\alpha_i+1+k)} + C_{(\alpha_i+1-k)} = 2\omega_i + x_1 + \sum_{j=0}^{\alpha_i} x_{n_i-2j} + \sum_{j=1}^{k} x_{2j-1} + \sum_{j=0}^{\alpha_i-k} x_{n_i-2j} & \text{when } n_i \text{ is odd.} \end{cases}$$

Due to the ordering in the alternating schedule, $\eta_{1,k}(\mathbf{x})$ is an increasing function of k:

$$\eta_{1,k+1}(\mathbf{x}) - \eta_{1,k}(\mathbf{x}) \;=\; \begin{cases} x_{2k+1} - x_{2k} & \text{when } n_i \text{ is even,} \\ x_{2k+2} - x_{2k+1} & \text{when } n_i \text{ is odd.} \end{cases}$$

We study two cases:

(i) If $\eta_{1,k}(\mathbf{x}) = C_{(\alpha_i+1+k)} + C_{(\alpha_i+1-k)} > 2\overline{C}_i(\sigma_i^*)$, then setting $k = \alpha_i$ in $\eta_{1,k+1}(\mathbf{x}_i)$, we get

$$C_{(n_i)} + C_{(1)} \leq MS + x_{n_i} + \omega_i$$

which (using (35)) implies

$$\left(\frac{1}{2}\eta_{1,k}(\mathbf{x}) - \overline{C}_i(\sigma_i^*)\right)^2 \;\leq\; \left(\frac{1}{2}(MS + x_{n_i} + \omega_i) - \omega_i - \frac{1}{2}(MS_i + x_{n_i-2})\right)^2 = \frac{1}{4}\left(x_{n_i} - \omega_i - x_{n_i-2}\right)^2.$$

(ii) If $\eta_{1,k}(\mathbf{x}_i) = C_{(\alpha_i+1+k)} + C_{(\alpha_i+1-k)} < 2\overline{C}_i(\sigma_i^*)$, then setting $k = 1$ in $\eta_{1,k+1}(\mathbf{x}_i)$, we get

$$C_{(\alpha_i+2)} + C_{(\alpha_i)} \geq MS - x_1.$$

To see why this is the case, consider the following bounding relationships on each partial summation of consecutive processing times in the alternating schedule as a proportion of the makespam:

$$\begin{aligned} x_{n_i} + x_{n_i-1} &\geq \tfrac{2}{n_i}(MS_i - \omega_i), \\ x_{n_i} + x_{n_i-1} + x_{n_i-2} + x_{n_i-3} &\geq \tfrac{4}{n_i}(MS_i - \omega_i), \\ x_{n_i} + x_{n_i-1} + x_{n_i-2} + x_{n_i-3} + x_{n_i-4} + x_{n_i-5} &\geq \tfrac{6}{n_i}(MS_i - \omega_i), \end{aligned}$$

and

$$\begin{aligned} x_{n_i} &\geq x_{n_i-1}, \\ x_{n_i} + x_{n_i-2} &\geq x_{n_i-1} + x_{n_i-3}, \\ x_{n_i} + x_{n_i-2} + x_{n_i-4} &\geq x_{n_i-1} + x_{n_i-3} + x_{n_i-5}. \end{aligned}$$

In the general case, for any $k = 1 \ldots \alpha_i + 1$ we have

$$\sum_{j=1}^{k} x_{n_i-j} \geq \frac{k}{n_i}(MS_i - \omega_i) \quad \text{and} \quad \sum_{j=1}^{k} x_{n_i-2(j-1)} \geq \sum_{j=1}^{k} x_{n_i-2(j-1)-1}.$$

This implies

$$\begin{aligned} x_{n_i} &\geq \tfrac{1}{n_i}(MS_i - \omega_i), \\ x_{n_i} + x_{n_i-2} &\geq \tfrac{2}{n_i}(MS_i - \omega_i), \\ x_{n_i} + x_{n_i-2} + x_{n_i-3} &\geq \tfrac{3}{n_i}(MS_i - \omega_i) \end{aligned}$$

and in general

$$x_{n_i} + \sum_{j=2}^{k} x_{n_i-j} \geq \frac{k}{n_i}(MS_i - \omega_i)$$

so that

$$C_{(\alpha_i+2)} + C_{(\alpha_i)} \geq x_{n_i-2\alpha_i} + x_{n_i-2(\alpha_i+1)} + \frac{2\alpha_i}{n_i}(MS_i - \omega_i) \geq \frac{2\alpha_i+2}{n_i}(MS_i - \omega_i) \geq (MS_i - \omega_i).$$

Therefore,

$$\left(\frac{1}{2}\eta_{1,k}(\mathbf{x}) - \overline{C}_i(\sigma_i^*)\right)^2 \;\leq\; \left(\frac{1}{2}(MS_i - \omega_i) - \omega_i - \frac{1}{2}MS_i + x_{n_i} + x_{n_i-3}\right)^2 \;=\; \frac{1}{4}\left(x_{n_i} + x_{n_i-3} - \omega_i\right)^2.$$

Using both cases (i) and (ii), we obtain

$$\frac{B_1(\overline{C}(\sigma^{AS}))}{B_0} \;\leq\; \frac{\max\left\{\frac{1}{4}\left(x_{n_i} - \omega_i - x_{n_i-2}\right)^2, \frac{1}{4}\left(x_{n_i} + x_{n_i-3} - \omega_i\right)^2\right\}}{B_0} \;=\; \frac{\kappa_i}{B_0}$$

where κ_i is a constant independent from n_i. Since

$$B_0 \geq \frac{1}{n_i}x_1^2 \sum_{k=1}^{\alpha_i}(2k-1)^2 = \frac{1}{n_i}x_1^2\left(\frac{3}{4}\alpha_i^3 + \frac{7}{3}\alpha_i\right) \geq \frac{x_1^2}{8}n_i^2,$$

we conclude that

$$\lim_{n_i \longrightarrow \infty} \frac{\sum_{k=0}^{\alpha_i}\left(\frac{1}{2}\eta_{1,k}(\mathbf{x}) - \overline{C}(\sigma^{AS})\right)^2}{\sum_{k=1}^{\alpha_i}(\eta_{0,k}(\mathbf{x}))^2} \;\leq\; \lim_{n_i \longrightarrow \infty} \left(\frac{\kappa_i}{x_1}\right)^2 \frac{8}{n_i^2} \;=\; 0.$$

■

Proof of Proposition 6. Let us define an order relationship $\prec$ over $\mathcal{M}$, such that for each $i, i' \in \mathcal{M}$, $i' \preceq i$ iff $x_{i,n_i} \geq x_{i',n_{i'}}$, and denote $i_m \preceq i_{m-1} \preceq \ldots \preceq i_1$. Then, it must be the case that

$$x_{i_1,n_i} = p_n. \tag{36}$$

Hence (25) is verified for i_1. To show that (25) is verified for all the remaining machines $i_2 \ldots i_m$, let i_a be the first machine for which (25) is not satisfied and let i_b be the machine where job $n-a+1$ is scheduled. Therefore,

$$i_a \prec i_{a-1} \preceq \ldots \preceq i_b \preceq \ldots \preceq i_1.$$

Then, by Hall and Kubiak (1991) we consider on the following partial sequences:

$$\mathcal{P}(\sigma_{i_b}) = (p_{n-b+1}, \alpha_i, \ldots, p_{n-a+1}) \qquad \text{and} \qquad \mathcal{P}(\sigma_{i_a}) = (\eta, \ldots),$$

with $\eta < p_{n-a+1} \leq p_{n-b+1}$ and $\alpha_i \leq p_{n-a+1}$. As before, $\mathcal{P}(\sigma)$ denote the order set of processing times of jobs in σ. We then construct other sequences π_{i_b} and π_{i_a} from σ_{i_b} and σ_{i_a} by permuting the positions of jobs η and p_{n-a+1} and keeping all the other jobs in the same position.

$$\mathcal{P}(\pi_{i_b}) = (p_{n-b+1}, \alpha_i, \ldots, \eta) \qquad \text{and} \qquad \mathcal{P}(\pi_{i_a}) = (p_{n-a+1}, \ldots).$$

Let $\delta = p_{n-a+1} - \eta$ and note that the CTV of sequence on machine i_a does not change under this permutation, as it constitutes a translation of all completion times at machine i_a:

$$CTV_{i_a}(\pi_{i_a}) = CTV_{i_a}(\sigma_{i_a}).$$

As far as machine i_b is concerned, we have:

$$\begin{cases} \overline{C}_{i_b}(\pi_{i_b}) = \overline{C}_{i_b}(\sigma_{i_b}) - \frac{\delta}{n_{i_b}}, \\ CTV_{i_b}(\pi_{i_b}) - CTV_{i_b}(\sigma_{i_b}) = \frac{\delta}{n_{i_b}} \left(2\overline{C}_{i_b}(\sigma_{i_b}) - 2MS_{i_b} + \frac{n_{i_b}-1}{n_{i_b}}\delta\right). \end{cases}$$

We proceed by showing that the aforementioned permutation from σ_{i_b} and σ_{i_a} to π_{i_b} and π_{i_a} gives rise to a decrease in the CTV at machine i_b. By contradiction, assume that $CTV_{i_b}(\pi_{i_b}) \geq CTV_{i_b}(\sigma_{i_b})$. Consequently,

$$\overline{C}_{i_b}(\sigma_{i_b}) \geq MS_{i_b} - \frac{n_{i_b}-1}{2n_{i_b}}\delta. \tag{37}$$

If σ_{i_b} is optimal for machine i_b (thus verifying Lemma 2), we invoke Proposition 3 and obtain

$$\overline{C}(\sigma_{i_b}) \leq \frac{n_{i_b}-1}{2n_{i_b}}MS_{i_b} + \frac{n_{i_b}+1}{2n_{i_b}}x_{i_b,n_{i_b}} - \frac{n_{i_b}-2}{2n_{i_b}}x_{i_b,n_{i_b}-1} + \frac{n_{i_b}+2}{4n_{i_b}}x_{i_b,n_{i_b}-2} + \frac{n_{i_b}-3}{2n_{i_b}}x_{i_b,n_{i_b}-3}. \tag{38}$$

Combining (37) and (38), we have

$$2(n_{i_b}+1)MS_{i_b} \leq 2(n_{i_b}+1)p_{n-b+1} - 2(2n_{i_b}-3)p_{n-a+1} + (3n_{i_b}-4)\alpha_i + 2(n_{i_b}-1)\eta. \tag{39}$$

Since $MS_{i_b} \geq p_{n-b+1} + p_{n-a+1} + \alpha_i$, then (39) becomes:

$$2(3n_{i_b}-2)p_{n-a+1} \leq (n_{i_b}-6)\alpha_i + 2(n_{i_b}-1)\eta.$$

Since $\alpha_i \leq p_{n-a+1}$ and $\eta < p_{n-a+1}$, then the last inequality implies that

$$(3n_{i_b}+4)p_{n-a+1} < 0$$

which is a contradiction. We conclude that $CTV_{i_b}(\pi_{i_b}) < CTV_{i_b}(\sigma_{i_b})$ which implies that setting $x_{i_1,i_a} = p_{n-a+1}$ is a better choice, when $x_{i_1,i_{a+1}} = p_{n-a+2}$. Therefore, since (36) is true, the dominance properties (25) are verified for all the machines $i \in \mathcal{M}$.

■

Proof of Proposition 7. Using the order $p_1 < p_2 < \ldots < p_n$ and the largest positions in each machine established in Proposition 6, we bound the makespan of each machine using the sequence of $n_i - 1$ smallest and largest elements:

$$\omega_i + p_{n-i+1} + \sum_{k=1}^{n_i-1} p_k \;\leq\; MS_i(\mathbf{x}_i) \;\leq\; \omega_i + p_{n-i+1} + \sum_{k=1}^{n_i-1} p_{n-k+1} \qquad \text{for } i \in \mathcal{M}. \tag{40}$$

From Proposition 3, we know that for each $i \in \mathcal{M}$:

$$MS_i(\mathbf{x}_i) \leq \omega_i + x_n + \frac{n-2}{2} x_{i,n-1} - \frac{n-6}{2} x_{i,n-2} + (n-3) x_{i,n-3},$$

and using Proposition 6, we get

$$MS_i(\mathbf{x}_i) \leq \omega_i + p_{n-i+1} + \frac{n-2}{2} x_{i,n-1} - \frac{n-6}{2} x_{i,n-2} + (n-3) x_{i,n-3}.$$

Combining these bounds on the makespan with (40), we obtain

$$MS_i(\mathbf{x}_i) \leq \omega_i + p_{n-i+1} + \min\left\{ \sum_{k=1}^{n_i-1} p_{n-k+1},\; \frac{n-2}{2} x_{i,n-1} - \frac{n-6}{2} x_{i,n-2} + (n-3) x_{i,n-3} \right\}.$$

■

Proof of Proposition 8. Assume that job indexes are ordered as $p_1 < p_2 < \ldots < p_n$ and order the indexes of machines is such a way as $n_m < n_{m-1} < \ldots < n_1$. Proposition 6 implies that

$$\sum_{i=1}^{m} x_{i,n_i} = \sum_{i=1}^{m} p_{n-i+1}$$

Note that job p_{n-m-1} must be the second largest job of some machine, as we know from Proposition 6 that $p_n \ldots p_{n-m}$ must all occupy the first positions. Then, by shifting the processing times by blocks of $n_1 \ldots n_m$, the second largest jobs of each machine cannot be smaller than

$$p_{n-m-1} + p_{n-m-n_1-1} + p_{n-m-n_1-n_2-1} + \ldots + p_{n-m-\sum_{s=1}^{i-1} n_s - 1}.$$

Therefore, the sum of the second largest jobs of each machine can be bounded to obtain (27).

■

References

Armony, M., R. Atar, and H. Honnappa. 2019. Asymptotically Optimal Appointment Schedules. *Mathematics of Operations Research* 44 (4): 1345–1380. pages 2

Arora, S., B. Bollobás, and L. Lovász. 2003. Proving integrality gaps without knowing the linear program. *Lecture Notes on Computer Science*:1–1. pages 2

Balouka, N., and I. Cohen. 2019. A robust optimization approach for the multi-mode resource-constrained project scheduling problem. *European Journal of Operational Research*. pages 1

Balseiro, S. R., D. B. Brown, and C. Chen. 2018. Static routing in stochastic scheduling: Performance guarantees and asymptotic optimality. *Operations Research* 66 (6): 1641–1660. pages 2

Beaumont, N. 1997. Scheduling staff using mixed integer programming. *European journal of Operational Research* 98 (3): 473–484. pages 2

Cai, X., and T. E. Cheng. 1998. Multi-machine scheduling with variance minimization. *Discrete applied mathematics* 84 (1-3): 55–70. pages 1

Castro, J., and S. Nasini. 2017. On geometrical properties of preconditioners in IPMs for classes of block-angular problems. *SIAM Journal on Optimization* 27 (3): 1666–1693. pages 18

Castro, J., S. Nasini, and F. Saldanha-da Gama. 2017. A cutting-plane approach for large-scale capacitated multi-period facility location using a specialized interior-point method. *Mathematical Programming* 163 (1-2): 411–444. pages 18

Chang, P.-Y., P. Damodaran*, and S. Melouk. 2004. Minimizing makespan on parallel batch processing machines. *International Journal of Production Research* 42 (19): 4211–4220. pages 4

Cheng, R., M. Gen, and T. Tozawa. 1995. Minmax earliness/tardiness scheduling in identical parallel machine system using genetic algorithms. *Computers & Industrial Engineering* 29 (1-4): 513–517. pages 2

Chicago Data Portal 2017. Performance Metrics - 311 Call Centers. data.cityofchicago.org/Administration-Finance/Performance-Metrics-311-Call-Center/kj88-xnxq. pages 15

Chlamtac, E., and M. Tulsiani. 2012. Convex relaxations and integrality gaps. In *Handbook on semidefinite, conic and polynomial optimization*, 139–169. Springer. pages 2

Chung, S., Y. Tai, and W. Pearn. 2009. Minimising makespan on parallel batch processing machines with non-identical ready time and arbitrary job sizes. *International Journal of Production Research* 47 (18): 5109–5128. pages 4

Codato, G., and M. Fischetti. 2006. Combinatorial Benders' cuts for mixed-integer linear programming. *Operations Research* 54 (4): 756–766. pages 18

Correa, J., A. Marchetti-Spaccamela, J. Matuschke, L. Stougie, O. Svensson, V. Verdugo, and J. Verschae. 2015. Strong LP formulations for scheduling splittable jobs on unrelated machines. *Mathematical Programming* 154 (1-2): 305–328. pages 2, 4

Eilon, S., and I. Chowdhury. 1977. Minimising waiting time variance in the single machine problem. *Management Science* 23 (6): 567–575. pages 1

Federgruen, A., and G. Mosheiov. 1996. Heuristics for multimachine scheduling problems with earliness and tardiness costs. *Management Science* 42 (11): 1544–1555. pages 2

Graham, R. L., E. L. Lawler, J. K. Lenstra, and A. R. Kan. 1979. Optimization and approximation in deterministic sequencing and scheduling: a survey. In *Annals of discrete mathematics*, Volume 5, 287–326. Elsevier. pages 2, 5

Green, L. V., S. Savin, and N. Savva. 2013. Nursevendor problem: Personnel staffing in the presence of endogenous absenteeism. *Management Science* 59 (10): 2237–2256. pages 3

Hall, N. G., and W. Kubiak. 1991. Proof of a conjecture of Schrage about the completion time variance problem. *Operations Research Letters* 10 (8): 467–472. pages 19, 22, 27

Jouglet, A., and J. Carlier. 2011. Dominance rules in combinatorial optimization problems. *European Journal of Operational Research* 212 (3): 433–444. pages 2

Kurpisz, A., M. Mastrolilli, C. Mathieu, T. Mömke, V. Verdugo, and A. Wiese. 2018. Semidefinite and linear programming integrality gaps for scheduling identical machines. *Mathematical Programming* 172 (1-2): 231–248. pages 2

Laures-Euskadi 2015. Modelo referencial de asignacin de tiempos de auxiliar de geriatra. http://www.lareseuskadi.org/uploads/media/MORETAG-Lares_Euskadi.pdf?fbclid=IwAR2V0tRgJycoUqIsz8eyScCBP9pmmWSYHgwEJHowo16sMfVA36Bt91Lb3pY. pages 12

Li, C.-L., and T. Cheng. 1994. The parallel machine min-max weighted absolute lateness scheduling problem. *Naval Research Logistics (NRL)* 41 (1): 33–46. pages 2

Li, L., D. J. Fonseca, and D.-S. Chen. 2006. Earliness–tardiness production planning for just-in-time manufacturing: A unifying approach by goal programming. *European Journal of Operational Research* 175 (1): 508–515. pages 1

Li, X., Y. Chen, and Y. Sun. 2010. Minimizing job completion time variance for service stability on identical parallel machines. *Computers & Industrial Engineering* 58 (4): 729–738. pages 1

Merten, A. G., and M. E. Muller. 1972. Variance minimization in single machine sequencing problems. *Management Science* 18 (9): 518–528. pages 1

Miltenburg, J. 1989. Level schedules for mixed-model assembly lines in just-in-time production systems. *Management Science* 35 (2): 192–207. pages 1

Muter, İ. 2020. Exact algorithms to minimize makespan on single and parallel batch processing machines. *European Journal of Operational Research*. pages 4

Oyetunji, E. 2009. Some common performance measures in scheduling problems. *Research Journal of Applied Sciences, Engineering and Technology* 1 (2): 6–9. pages 1

Schrage, L. 1975. Minimizing the time-in-system variance for a finite jobset. *Management Science* 21 (5): 540–543. pages 19

Srirangacharyulu, B., and G. Srinivasan. 2010. Completion time variance minimization in single machine and multi-machine systems. *Computers & Operations Research* 37 (1): 62–71. pages 1

T'kindt, V., and J.-C. Billaut. 2006. *Multicriteria scheduling: theory, models and algorithms.* Springer Science & Business Media. pages 2

Van de Vonder, S., E. Demeulemeester, and W. Herroelen. 2008. Proactive heuristic procedures for robust project scheduling: An experimental analysis. *European Journal of Operational Research* 189 (3): 723–733. pages 1

Verschae, J., and A. Wiese. 2014. On the configuration-LP for scheduling on unrelated machines. *Journal of Scheduling* 17 (4): 371–383. pages 2

Viswanathkumar, G., and G. Srinivasan. 2003. A branch and bound algorithm to minimize completion time variance on a single processor. *Computers & Operations Research* 30 (8): 1135–1150. pages 1