Valid constraints for time-indexed formulations of job scheduling problems with distinct time windows and sequence-dependent setup times

Full Paper

Bruno Ferreira Rosa
Federal Center of Technological Education of Minas Gerais
Divinópolis, MG, Brazil
brunorosa@cefetmg.br

Marcone Jamilson Freitas Souza
Department of Computing, Federal University of Ouro Preto
Ouro Preto, MG, Brazil
marcone@ufop.edu.br

Sérgio Ricardo de Souza
Federal Center of Technological Education of Minas Gerais
Belo Horizonte, MG, Brazil
sergio@dppg.cefetmg.br

Zacharie Ales
ENSTA ParisTech/UMA
Paris, France
zacharie.ales@ensta-paristech.fr

Philippe Yves Paul Michelon
University of Avignon
Avignon, France
philippe.michelon@univ-avignon.fr

ABSTRACT

This paper addresses the single machine scheduling problem with distinct time windows, sequence-dependent setup times (SMSPETP) which consists in minimizing the total weighted earliness and tardiness of a set of jobs. We propose a time-indexed mathematical formulation for representing the problem, new valid constraints families for this formulation, as well as separation algorithms. Computational experiments show that the use of these algorithms in a cutting-plane enable to significantly improve the linear relaxation.

1 INTRODUCTION

This paper addresses the single machine scheduling problem with distinct time windows and sequence-dependent setup times. Such problem consists of determining the time at which jobs must be performed in order to minimize the weighted sum of earliness and tardiness penalties, and is hereafter denoted by SMSPETP.

The SMSPETP is a difficult problem which has numerous applications, such as Just-in-Time manufacturing, chemical processing, video on demand services, among others. As a consequence, many resolution algorithms have been introduced to solve this problem [3, 5]. Nevertheless, the job scheduling problem with the characteristics considered in this work has not received the deserved attention. The SMSPETP has mainly been treated by heuristic procedures that divide the problem into two subproblems: (i) job sequencing, and (ii) determining the optimal time for completion of each job in a given sequence. This work tackles the SMSPETP from a perspective not yet considered in the literature i.e., with a cutting plane algorithm.

The SMSPETP has the following characteristics:

- A single machine must process a set I of n jobs;
- The machine can perform only one job at a time and, once the process is initiated, it cannot be interrupted;
- All jobs are available for processing starting from date 0;
- Between two consecutive jobs x and $y \in I$, a setup time of S_{xy} is required. It is assumed that the time for setting up the machine in order to process the first job in the sequence is equal to 0;
- Idle time between the execution of two consecutive jobs is allowed.
- For each job $x \in I$, there is a processing time P_x and a time window $[E_x, T_x]$ in which the job x should preferably be completed. E_x indicates the earliest due date, and T_x is the tardiest due date;
- If job x is completed before E_x, then there is a cost of α_x per unit of earliness time. In the case that the job is completed after T_x, there is a cost of β_x per unit of tardiness time. Jobs completed within their time windows do not incur costs;

The objective of the problem is to determine the starting dates of the jobs, so that the weighted sum of their earliness and tardiness is minimized, i.e.,

$$\min \sum_{x \in I} (\alpha_x e_x + \beta_x t_x), \tag{1}$$

where C_x represents the completion time of job $x \in I$ and $e_x = \max(0,\ E_x - C_x)$ and $t_x = \max(0, C_x - T_x)$ represent the earliness and tardiness times of x, respectively.

In this paper, a time-indexed formulation for representing the SMSPETP is presented. In addition, five families of valid constraints for time-indexed SMSPETP formulations are proposed in order to obtain better lower bounds.

The rest of this article is organized as follows. The time-indexed formulation for the SMSPETP is presented in Section 2, while the five families of valid constraints for time-indexed SMSPETP formulations are showed in Section 3. Section 4 proposes separations algorithms for these families of constraints. Section 5 presents and discusses the computational results. Finally, Section 6 concludes this work.

2 THE PROPOSED TIME-INDEXED FORMULATION

In [6, 7] were introduced time-indexed formulations of the single machine scheduling problem with distinct deadlines and no setup times. We adapt these formulations and use the valid constraints of [1] in order to represent the SMSPETP.

Let $H_x = \{s_x^{LB}, s_x^{LB}+1, \ldots, s_x^{UB}\}$ be the set of possible starting dates of job $x \in I$. Let l_{xh} be decision variables such that $\forall\, x \in I$ and $\forall\, h \in H_x$,

$$l_{xh} = \begin{cases} 1, & \text{if job } x \text{ begins at date } h; \\ 0, & \text{otherwise.} \end{cases}$$

In the rest of this work the following notations are used: $\lfloor \lambda \rfloor_x = \max(s_x^{LB}, \lambda)$ and $\lceil \lambda \rceil_x = \min(\lambda, s_x^{UB})$, for all job $x \in I$ and for all number $\lambda \in R$.

As introduced in [7], the cost incurred by the earliness or tardiness of a job $x \in I$ started at date h can be determined by the function

$$g_x(h) = \alpha_x \cdot \max(E_x - h - P_x, 0) + \beta_x \cdot \max(h + P_x - T_x, 0),\ \forall\, h \in H_x \quad (2)$$

Therefore, a time-indexed formulation for the SMSPETP, denoted by TIF, is given by

$$\text{(TIF)} \qquad \min \sum_{x\in I} \sum_{h \in H_x} g_x(h) \cdot l_{xh}$$

$$\text{s.t.} \quad \sum_{h\in H_x} l_{xh} = 1, \quad \forall x \in I \quad (3)$$

$$\sum_{k=\lfloor h-P_x-S_{xy}+1\rfloor_x}^{\lceil h \rceil_x} l_{xk} + \sum_{k=\lfloor h-P_y-S_{yx}+1\rfloor_y}^{\lceil h \rceil_y} l_{yk} \le 1, \quad \forall x, y \in I, x \ne y, \quad \forall h \in H_x \cup H_y \quad (4)$$

$$l_{xh} \in \{0,\ 1\}, \quad \forall x \in I, \forall h \in H_x \quad (5)$$

The objective function seeks to minimize the weighted sum of the earliness and tardiness. Constraints (3) assure that each job will be executed only once. Constraints (4) ensure that there is sufficient time to execute a job and prepare the machine before starting the next job. Note that Constraints (4) assume the validy of the triangle inequality given by

$$S_{xy} \le S_{xz} + P_z + S_{zy}, \quad \forall\ x,\ y,\ z \in I,\ x \ne y,\ x \ne z \text{ and } y \ne z. \quad (6)$$

3 NEW VALID CONSTRAINTS

This section introduces new families of valid constraints for time-indexed formulations to the SMSPETP.

Before presenting the new valid constraints, it is observed that the constraints given by Proposition 3.1 of [1] are also valid for the time-indexed formulations of SMSPETP. These constraints are used to prove the validity of the first family of valid constraints.

Proposition 3.1 ([1]). *Given a subset $I' \subseteq I$ such that $|I'| \ge 2$, for all $h \in \bigcup_{x\in I'} H_x$, we have*

$$\sum_{x\in I'} \sum_{k=\left\lfloor h-P_x-\min\limits_{y\in I'\setminus\{x\}} S_{xy}+1\right\rfloor_x}^{\lceil h \rceil_x} l_{xk} \le 1. \quad (7)$$

Note that Constraints (4) are obtained from Constraints (7) by considering only the subsets $I' \subset I$ such that $|I'| = 2$.

The first family of valid constraints is inspired by [6]. In this work, the authors propose the set of Constraints (8) for time-indexed formulations of scheduling problems without setup time between jobs:

$$\sum_{k=\lfloor h-P_x+1\rfloor_x}^{\lceil h+\Delta-1 \rceil_x} l_{xk} + \sum_{y\in I\setminus\{x\}: P_y \ge \Delta} \sum_{k=\lfloor h-P_y+\Delta\rfloor_y}^{\lceil h \rceil_y} l_{yk} \le 1, \quad \forall x \in I, \quad \forall h \in \bigcup_{y\in I\setminus\{x\}} H_y,\ \forall \Delta \in \{2, 3, \ldots, \max_{y\in I\setminus\{x\}} P_y\} \quad (8)$$

Proposition 3.2 generalizes Constraints (8) to scheduling problems with setup times between jobs. The family of constraints that satisfies Proposition 3.2 is named here "Family 1".

Proposition 3.2 (Family 1). *Let $I' \subseteq I$ be a subset of jobs such that $|I'| \ge 2$. Given a job $x \in I'$, for all $h \in \bigcup_{y\in I'\setminus\{x\}} H_y$ and all $\Delta \in \{2 - P_x - \min_{y\in I'\setminus\{x\}} S_{xy}, \ldots, \max_{y\in I'\setminus\{x\}} (P_y + S_{yx})\}$, we have*

$$\underbrace{\sum_{k=\lfloor h-P_x-S_x^{min}+1\rfloor_x}^{\lceil h+\Delta-1 \rceil_x} l_{xk}}_{\epsilon_x} + \sum_{y\in I^*} \underbrace{\sum_{k=\lfloor h-P_y-S_y^{min}+\Delta\rfloor_y}^{\lceil h \rceil_y} l_{yk}}_{\epsilon_y} \le 1, \quad (9)$$

where:

- $I^* = \{y \in I' \setminus \{x\} \mid P_y + S_{yx} \ge \Delta\}$,
- $S_x^{min} = \min_{y\in I^*} S_{xy}$ *and*
- $S_y^{min} = \min\left(S_{yx}, \Delta - 1 + \min_{z\in I^*\setminus\{y\}} S_{yz}\right)$ *for all* $y \in I^*$ $\left(\textit{if } I^* = \{y\},\ \textit{then } S_y^{min} = S_{yx}\right)$.

Proof. As job x must be processed once, we have $\varepsilon_x \le 1$. Moreover, we have from Proposition 3.1 that $\sum_{y\in I^*} \epsilon_y \le 1$. Suppose there is a feasible scheduling π of I in which $\varepsilon_x = 1$ and $\sum_{y\in I^*} \epsilon_y = 1$ for a given h' and a given Δ'. Thus, there is a job $y' \in I^*$ such that $\epsilon_{y'} = 1$. Consequently, $h' - P_x - S_{xy'} + 1 \le s_x^\pi \le h' + \Delta' - 1$ and $h' - P_{y'} - S_{y'x} + \Delta' \le s_{y'}^\pi \le h'$, where s_x^π is the starting date of job x in scheduling π, i.e., there is overlap between jobs x and y' in scheduling π. □

Note that Constraints (7) are contained in Family 1. In fact, they are obtained by only setting $\Delta = 1$ in Family 1.

The following lemma provides a new set of valid constraints.

Lemma 3.3. *For any subset $I' \subseteq I$, we have*

$$\sum_{x\in I'} \underbrace{\left(\sum_{k=\lfloor h \rfloor_x}^{\lceil h+\min\limits_{y\in I'\setminus\{x\}}(P_y+S_{yx})-1 \rceil_x} l_{xk} \right)}_{\epsilon_x} \le 1, \quad \forall\, h \in \bigcup_{x\in I'} H_x. \quad (10)$$

Proof. As every job must be processed once, we have $\epsilon_x \le 1$, $\forall x \in I'$ and $\forall h \in \bigcup_{x\in I'} H_x$. Suppose there is a feasible scheduling π of I such that $\sum_{x\in I'} \epsilon_x > 1$ for a given h'. Therefore, there are two jobs $x_1, x_2 \in I'$ such that $\epsilon_{x_1} = \epsilon_{x_2} = 1$. Consequently, $h' \le s_{x_1}^\pi \le h' + P_{x_2} + S_{x_2,x_1} - 1$ and $h' \le s_{x_2}^\pi \le h' + P_{x_1} + S_{x_1,x_2} - 1$, where s_x^π is the starting date of job x in scheduling π, that is, the machine performs jobs x_1 and x_2 in scheduling π simultaneously. This contradicts the fact that π is a feasible scheduling of I. □

Proposition 3.4 provides another family of valid constraints, named "Family 2". This family contains Constraints (10).

Proposition 3.4 (Family 2). *Let $I' \subseteq I$ be a subset of jobs such that $|I'| \ge 2$. Given a job $x \in I'$, for all $h \in \bigcup_{y\in I'\setminus\{x\}} H_y$ and all $\Delta \in \{2 - \min_{y\in I^*} (P_y + S_{yx}), \ldots, P_x + \max_{y\in I'\setminus\{x\}} S_{xy}\}$, we have:*

$$\underbrace{\sum_{k=\lfloor h-\Delta+1\rfloor_x}^{\lceil h+PS_x^{min}-1 \rceil_x} l_{xk}}_{\epsilon_x} + \sum_{y\in I^*} \underbrace{\sum_{k=\lfloor h \rfloor_y}^{\lceil h+PS_y^{min}-\Delta \rceil_y} l_{yk}}_{\epsilon_y} \le 1, \quad (11)$$

where:

- $I^* = \{y \in I' \setminus \{x\} | P_x + S_{xy} \ge \Delta\}$,
- $PS_x^{min} = \min_{y\in I^*} (P_y + S_{yx})$ *and*
- $PS_y^{min} = P_x + S_{xy}$, *if* $I^* = \{y\}$, *or* $PS_y^{min} = \min\left(P_x + S_{xy}, \Delta - 1 + \min_{z\in I^*\setminus\{y\}} (P_z + S_{zy})\right)$ *for all* $y \in I^*$, *otherwise.*

Proof. Since job x must be performed only once, we have $\varepsilon_x \leq 1$. Besides it, from Constraints (10) we have $\sum_{y \in I^*} \epsilon_y \leq 1$. Suppose there is a schedule π from I such that $\varepsilon_x = 1$ and $\sum_{y \in I^*} \epsilon_y = 1$ to a given date h' and a given Δ'. Therefore, there is $y' \in I^*$ such that $\epsilon_{y'} = 1$. Consequently, $h' - \Delta' + 1 \leq s_x^\pi \leq h' + P_{y'} + S_{y'x} - 1$ and $h' \leq s_{y'}^\pi \leq h' + P_x + S_{xy'} - \Delta'$, where s_x^π is the starting date of job x in scheduling π, that is, the machine performs jobs x and y' in schedule π simultaneously. This contradicts the fact that π is a feasible schedule of I. □

Constraints (10) are obtained from Family 2 when considering $\Delta = 1$. Propositions 3.5, 3.6 and 3.7 provide three more families of valid constraints, which will be named "Family 3", "Family 4" and "Family 5" respectively.

Proposition 3.5 (Family 3). *For any subset $I' \subseteq I$ such that $|I'| \geq 2$, we have:*

$$\underbrace{\sum_{x \in I'} \sum_{k=s_x^{LB}}^{\left\lceil \min\limits_{y \in I' \setminus \{x\}} (s_y^{LB} + P_y + S_{yx}) - 1 \right\rceil_x} l_{xk}}_{\epsilon_x} \leq 1. \tag{12}$$

Proof. According to what has already been discussed, $\epsilon_x \leq 1, \forall x \in I'$. Suppose there is a schedule π of I such that $\sum_{x \in I'} \epsilon_x > 1$. Therefore, there are two jobs x_1,$x_2 \in I'$ such that $\epsilon_{x_1} = \epsilon_{x_2} = 1$. Consequently, $s_{x_1}^{LB} \leq s_{x_1}^\pi \leq s_{x_2}^{LB} + P_{x_2} + S_{x_2,x_1} - 1$ and $s_{x_2}^{LB} \leq s_{x_2}^\pi \leq s_{x_1}^{LB} + P_{x_1} + S_{x_1,x_2} - 1$, where s_x^π is the starting date of job x in scheduling π, i.e., the machine performs jobs x_1 and x_2 in scheduling π simultaneously. This contradicts the fact that π is a feasible scheduling of I. □

Proposition 3.6 (Family 4). *Given a subset of jobs $I' \subseteq I$ such that $|I'| \geq 2$, if $TPT_{\min}^{I'}$ denotes the lowest total time required to process all jobs in I' and $s_{I'}^{LB} = \min_{x \in I'} s_x^{LB}$, then*

$$\underbrace{\sum_{x \in I'} \sum_{h=\left\lfloor s_{I' \setminus \{x\}}^{LB} + TPT_{\min}^{I' \setminus \{x\}} + \min\limits_{y \in I \setminus \{x\}} S_{yx} \right\rfloor_x}^{s_x^{UB}} l_{xh}}_{\epsilon} \quad \geq \quad 1. \tag{13}$$

Proof. Suppose there is a feasible scheduling π of I such that ϵ is equal to 0. Let s_x^π be the starting date of job x in scheduling π and $x' \in I'$ be the last job processed in π. Thus, $s_{x'}^\pi < s_{I' \setminus \{x\}}^{LB} + TPT_{\min}^{I' \setminus \{x'\}} + \min_{y \in I \setminus \{x'\}} S_{yx'}$ and there is a scheduling of $I' \setminus \{x'\}$ whose total processing time is lower than $TPT_{min}^{I' \setminus \{x'\}}$. This contradicts the fact that $TPT_{\min}^{I' \setminus \{x'\}}$ is the lowest total time required to process all jobs in $I' \setminus \{x'\}$. □

Proposition 3.7 (Family 5). *Given a pair of distinct jobs x and y in I, for all $h \in \left\{ \max\left(s_x^{LB}, s_y^{LB}\right), \ldots, \min\left(s_x^{UB}, s_y^{LB} + P_y + S_{yx} - 1\right) \right\} \cap H_x$, we have*

$$\underbrace{\sum_{k=h+P_x+S_{xy}}^{s_y^{UB}} l_{yk}}_{\epsilon} \geq l_{xh}. \tag{14}$$

Proof. Suppose there is a feasible scheduling π of I such that $\epsilon < l_{xh}$ for a given h (i.e., such that $l_{xh} = 1$ and $\epsilon = 0$). So, $s_x^\pi = h$ and $s_y^{LB} \leq s_y^\pi < h + P_x + S_{xy}$, where s_x^π denotes the starting date of job x in scheduling π. Since $s_y^{LB} \leq h \leq s_y^{LB} + P_y + S_{yx} - 1$, it follows that $s_x^\pi + P_x + S_{xy} > s_y^\pi$ and $s_y^\pi + P_y + S_{yx} > s_x^\pi$, contradicting the fact that π is a feasible scheduling of I. □

4 SEPARATION ALGORITHMS FOR THE FAMILIES OF PROPOSED CONSTRAINTS

Except Family 5, all families of constraints proposed in Section 3 have an exponential number of constraints (2^n or greater). This fact makes it impossible to fully include these families in the formulations. However, they can be used in cutting-plane algorithms [8]. In short, cutting-plane algorithms are procedures that start from the solution of the linear relaxation of a formulation in which a limited number of constraints are considered and iteratively adds new valid constraints to the problem and solve it, until one stopping criterion is satisfied.

Let *PPM* be the mathematical programming problem based on time-indexed variables that is updated iteratively in a given cutting-plane algorithm. Consider that $l^\star$ represents an optimal solution of the linear relaxation of the current *PPM*. Note that $l^\star$ consists of an array of values assigned to the variables l_{xh}, $\forall x \in I$ and $\forall h \in H_x$. Due to the large number of constraints in Families 1–4, the simple fact of checking which constraints are violated by $l^\star$ is still an impractical process. The problem of finding, in a set of constraints, those that are violated by $l^\star$ is called a "separation problem".

The separation problem associated with Family 5 is solved exactly by checking all constraints, one by one. On the other hand, the separation problems of Families 1–4 are solved heuristically. Moreover, the separation algorithms seek only the constraint which are the most violated by $l^\star$. Constraints whose violation by $l^\star$ is small are discarded. A constraint of type $A \times l \leq b$ is violated by $l^\star$ for at least $\delta > 0$ units if $A \times l^\star \geq b + \delta$.

The algorithms proposed to solve the separation problems associated with the families of constraints presented in Section 3 are described in the following subsections. Let δ represents the minimum violation accepted. Given a solution $l^\star$ for the current *PPM*, let $h_x^{\min} = \min\{h \in H_x : l_{xh}^\star > 0\}$ and $h_x^{\max} = \max\{h \in H_x : l_{xh}^\star > 0\}$, $\forall x \in I$. Furthermore, for each job $x \in I$, let $I_x = \{y \in I \setminus \{x\} : h_y^{\max} + P_y + S_{yx} > h_x^{\min}$ or $h_x^{\max} + P_x + S_{xy} > h_y^{\min}\}$.

4.1 Separation Heuristic for Family 1

The proposed separation heuristic algorithm for Family 1 is described in Algorithm 1. In this algorithm, Ω_1 represents the set of constraints violated by $l^\star$. $lhs_{x,h,I',\Delta}(l^\star)$ represents the numerical value of the expression $\varepsilon_x + \sum_{y \in I^*} \epsilon_y$ related to Proposition 3.2 applied to $l^\star$, for the respective x, h, I' and Δ.

The jobs are sorted in ascending order by the values of $h_x^{\min}$. The maximum number of constraints returned by the separation heuristic of Family 1 is given by $\sum_{x \in I}(h_x^{\max} - h_x^{\min} + 1)$.

4.2 Separation Heuristic for Family 2

Let Ω_2 be a subset of Family 2 composed of constraints which are violated by $l^\star$. If $lhs_{x,h,I',\Delta}(l^\star)$ represents the numerical value of the expression $\varepsilon_x + \sum_{y \in I^*} \epsilon_y$ of Proposition 3.4 applied to $l^\star$, then the proposed heuristic algorithm for the separation problem of Family 2 is analogous to Algorithm 1. The only differences are the range of Δ and the function $lhs_{x,h,I',\Delta}(l^\star)$, which, in this case, are based on Proposition 3.4.

```
Input: I; h_x^max, h_x^min, I_x, lhs_{x,h,I',Δ}(l★) ∀x ∈ I; l★; δ ∈ R.
Ω1 ← ∅;
for x ∈ I do
    for h = h_x^max, h_x^max − 1, ··· , h_x^min do
        I' ← {x};
        lhs0 ← −∞;
        Update ← FALSE;
        while I' ≠ I_x ∪ {x} do
            y* ← −1;
            lhs* ← −∞;
            for y ∈ I_x \ I' do
                I' ← I' ∪ {y};
                for Δ = P_y + S_yx, P_y + S_yx − 1, ··· , 2 −
                 P_x − min_{z∈I\{x}} S_xz do
                    if lhs_{x,h,I',Δ}(l★) ≥ 1 + δ then
                        Ω1 = Ω1 ∪ {lhs_{x,h,I',Δ}(l) ≤ 1};
                        Update ← TRUE;
                        Exit the current loop;
                    if lhs_{x,h,I',Δ}(l★) > lhs* then
                        lhs* ← lhs_{x,h,I',Δ}(l★);
                        y* ← y;
                if Update = TRUE then
                    Exit the current loop;
                else
                    I' ← I' \ {y};
            if Update = TRUE then
                Exit the current loop;
            if lhs* > lhs0 then
                lhs0 ← lhs*;
            else
                Exit the current loop;
            I' ← I' ∪ {y*};
Return Ω1;
```

Algorithm 1: Separation Heuristic for Family 1.

```
Input: I; I_x, lhs_I'(l★) ∀x ∈ I; l★; δ ∈ R.
Ω3 ← ∅;
for x ∈ I do
    I' ← {x};
    lhs0 ← −∞;
    Update ← FALSE;
    while I' ≠ I_x ∪ {x} do
        y* ← −1;
        lhs* ← −∞;
        for y ∈ I_x \ I' do
            I' ← I' ∪ {y};
            if lhs_I'(l★) ≥ 1 + δ then
                Ω3 = Ω3 ∪ {lhs_I'(l) ≤ 1};
                Update ← TRUE;
                Exit the current loop;
            if lhs_I'(l★) > lhs* then
                lhs* ← lhs_I'(l★);
                y* ← y;
            I' ← I' \ {y};
        if Update = TRUE then
            Exit the current loop;
        if lhs* > lhs0 then
            lhs0 ← lhs*;
        else
            Exit the current loop;
        I' ← I' ∪ {y*};
Return Ω3;
```

Algorithm 2: Separation Heuristic for Family 3.

4.3 Separation Heuristic for Family 3

The proposed heuristic algorithm for the separation problem of Family 3 is detailed in Algorithm 2. Ω_3 represents the set of constraints violated by $l^\star$ found by this algorithm. Let $lhs_{I'}(l^\star)$ represents the numerical value of the expression $\sum_{x \in I'} \epsilon_x$ of Proposition 3.5 applied to $l^\star$, for the respective subset $I' \subseteq I$.

As in the separation heuristics of Families 1 and 2, the order of investigation of the jobs $x \in I$ is always given in the increasing order of $h_x^{\min}$. The maximum number of constraints returned by the separation heuristic of Family 3 is equal to n.

4.4 Separation Heuristic for Family 4

Before presenting the proposed separation for Family 4, it is observed Constraint (13) of Proposition 3.6 is equivalent to Constraint (15) for all subset $I' \subseteq I$.

$$\underbrace{\sum_{x \in I'} \sum_{h=s_x^{LB}}^{\left\lceil s_{I' \setminus \{x\}}^{LB} + TPT_{\min}^{I' \setminus \{x\}} + \min_{y \in I \setminus \{x\}} S_{yx} - 1 \right\rceil_x} l_{xh}}_{\epsilon'_{I'}} - |I'| \leq -1. \qquad (15)$$

Let Ω_4 be a subset of Family 4 composed of constraints that are violated by $l^\star$. Let $lhs_{I'}(l^\star)$ be the numerical value of expression $\epsilon'_{I'} - |I'|$ of Constraint (15) applied to $l^\star$, for the respective subset $I' \subseteq I$.

The heuristic algorithm proposed for the separation problem of Family 4 is similar to that of Family 3. The only difference is in the function $lhs_{I'}(l^\star)$, which, in this case, is based on Constraint (15). In addition, instead of the exact value of $TPT_{\min}^{I'}$, a lower bound is used for that value. The lower bound used is provided by Corollary 4.1, which follows from the results proposed in [4].

Corollary 4.1. *For every subset $I' \subseteq I$, the shortest total time required to perform all jobs of I', that is $TPT_{min}^{I'}$, is such that*

$$TPT_{\min}^{I'} \geq \sum_{x \in I'} P_x + \max\left(\sum_{x \in I'} \min_{y \in I' \setminus \{x\}} S_{yx} - \max_{x \in I'} \min_{y \in I' \setminus \{x\}} S_{yx}, \right.$$
$$\left. \sum_{x \in I'} \min_{y \in I' \setminus \{x\}} S_{xy} - \max_{x \in I'} \min_{y \in I' \setminus \{x\}} S_{xy} \right).$$

4.5 Separation Algorithm for Family 5

The separation of Family 5 is solved exactly.

The proposed algorithm for the separation problem of Family 5 is detailed in Algorithm 3. Ω_5 represents the set of constraints violated by $l^\star$ found by this algorithm.

5 COMPUTATIONAL RESULTS

This Section presents the computational results obtained with the time-indexed formulation for the SMSPETP presented in Section 2, as well as with the different families of constraints proposed in Section 3. The separation algorithms described in Section 4 are used in a cutting plane framework in order to experiment how much they enable to improve the linear relaxation.

The mathematical formulations were implemented and solved through the C++ Concert Technology tool and the IBM ILOG CPLEX Optimization Studio 12.6.2 solver. The separation heuristics used for testing the proposed families of constraints were

Input: I; s_x^{LB}, s_x^{UB} $\forall x \in I$; $l^\star$; $\delta \in R$.
$\Omega_5 \leftarrow \emptyset$;
for $x \in I$ **do**
  for $y \in I \setminus \{x\}$ **do**
    for $h = \max(s_x^{LB}, s_y^{LB}), \max(s_x^{LB}, s_y^{LB}) + 1, \cdots, \min(s_x^{UB}, s_y^{LB} + P_y + S_{yx} - 1)$ **do**
      if $\sum_{k=h+P_x+S_{xy}}^{s_y^{UB}} l^\star_{yk} \leq l^\star_{xh} - \delta$ **then**
        $\Omega_5 = \Omega_5 \cup \{ \sum_{k=h+P_x+S_{xy}}^{s_y^{UB}} l_{yk} \geq l_{xh} \}$;
Return Ω_5;

Algorithm 3: Separation Algorithm for Family 5.

also implemented in C++ language. The experiments were realized on a computer Intel® Xeon(R) CPU E5620 @ 2.40GHz × 16, with 48 GB of RAM and CentOS Linux 7 operation system. CPLEX was configured to use only one thread and the other parameters were not changed. In addition, the algorithms were not optimized for multiprocessing.

A set of instances of [5], involving up to 20 jobs and satisfying the triangle inequality, was used in order to test the proposed formulations. This set contains 16 instances of each value of n. For each job $x \in I$, the bounds s_x^{UB} and s_x^{LB} used for determining the parameter values of each mathematical formulation are the same than in [4].

The cutting-plane algorithm described in Algorithm 4 was used in order to obtain lower bounds to the SMSPETP. The strategy that was used is based on the Variable Neighborhood Descent – VND [2] procedure. It uses a subsequencing of m separation algorithms proposed in Section 4, where $1 \leq m \leq 5$,.

$PPM \leftarrow PPM_0$;
$l^\star \leftarrow$ solution of PPM;
$\delta \leftarrow 0.8$;
while $\delta \geq 0.1$ **do**
  $i \leftarrow 1$;
  while $i \leq m$ **do**
    Solve the separation problem related to the i-th family of constraints for $l^\star$ and δ;
    if *there are constraints that are violated by* $l^\star$ **then**
      Add these constraints to the current PPM;
      $l^\star \leftarrow$ solution of the current PPM;
      Eliminate from the current PPM the constraints satisfied by $l^\star$ with non-zero slack;
      $i \leftarrow 1$;
    else
      $i \leftarrow i + 1$;
  $\delta \leftarrow \delta \div 2$;
Return $l^\star$;

Algorithm 4: Lower Bound obtained with m families of constraints.

In Algorithm 4, the initial PPM is provided by the PPM_0 formulation, defined by Equations (16)–(18).

$$(PPM_0) \quad \min \sum_{x \in I} \sum_{h \in H_x} g_x(h) \cdot l_{xh} \tag{16}$$

$$\text{s.t.} \quad \sum_{h \in H_x} l_{xh} = 1 \quad \forall x \in I \tag{17}$$

$$l_{xh} \in [0, 1] \quad \forall x \in I \text{ and } \forall h \in H_x \tag{18}$$

Equation (16) represents the objective function of SMSPETP. Constraints (17) ensure that each job must be executed once.

Given an instance of the problem, the gap of a given lower bound LB with respect to a given integer solution value $f^\star$ is determined by Equation (19):

$$gap = \frac{f^\star - LB}{f^\star} \times 100. \tag{19}$$

The lower the value of the gap, the better the lower bound LB is. We consider the best integer solutions from [5] to compute the gaps.

The results are reported in Table 1. In this table, the first column indicates the number of jobs of each set consisting of 16 instances. Columns "TIF" present the results using the linear relaxation of the proposed time-indexed formulation. Columns "Family 1", "Family 2", …, "Family 5" report the results by applying Algorithm 4 with the corresponding separation algorithm. Columns "Family 1–5" show the results by applying the Algorithm 4 with the five proposed separation algorithms in this order: Families 1, 2, 4, 3 and 5. For each set of instances, columns "$\overline{gap}$" and "$\overline{\text{time}}$" show, respectively, the average gap of the lower bounds (in %) and the average time, in seconds, required for each strategy over the 16 corresponding instances.

According to Table 1, the smallest average gaps obtained with only one family of constraints are Family 1, followed by Families 2, 4, 3 and 5, in this order (this justifies the choice of this sequence of separation algorithms when using all the constraint families). The difference between the average gaps of the lower bounds obtained with Family 1 and the average gaps obtained with Family 2 is relevant. The same happens with the difference between the average gaps of the lower bounds constructed with Families 2 and 4. The larger average times were also observed when using Family 1, followed by the average times required with Family 2. The average times required by Families 3, 4 and 5 were less than 2 seconds. However, the average gaps of the lower bounds constructed with these families of constraints were greater than or equal to 72.00 %.

Also according to Table 1, the average times required to obtain the lower bounds with the Families 1–5 were always higher than the average time required for solving the linear relaxation of the TIF formulation. However, the average gaps of the lower bounds resulting from the application of Algorithm 4 are significantly lower than the average gaps obtained with linear relaxation. The lower gaps of the average gaps obtained with the linear relaxations of the TIF formulation are greater than 37%, while the average gaps obtained by Families 1–5 are less than 6%. The average gap of the lower bounds obtained with the families 1–5 for the instances with 6 jobs are null, that is, the Algorithm 4 has found the optimal whole solutions of these problems. Although it is not shown in Table 1, the Algorithm 4 has found the optimal integer solutions of a total of 87 instances, among them an instance with 20 jobs.

6 CONCLUSIONS

In this work a time-indexed formulation, named TIF, for solving the Single Machine Scheduling Problem with distinct time windows and sequence-dependent setup times (SMSPETP) is proposed. Five new families of valid constraints for time-indexed formulations as well as separation algorithms for these families are also introduced.

Table 1: Results obtained when applying the Algorithm 1 in the instances.

n	TIF		Family 1		Family 2		Family 3		Family 4		Family 5		Families 1-5	
	$\overline{gap}$ (%)	time (s)	$\overline{gap}$ (%)	time (s)	$\overline{gap}$ (%)	time (s)	$\overline{gap}$ (%)	time (s)	$\overline{gap}$ (%)	time (s)	$\overline{gap}$ (%)	time (s)	$\overline{gap}$ (%)	time (s)
06	37.85	0.25	0.06	0.68	1.59	1.49	84.90	0.06	72.00	0.08	87.83	0.08	0.00	0.65
07	49.03	0.47	0.17	2.95	4.26	5.31	83.77	0.10	74.81	0.10	87.87	0.09	0.02	2.79
08	55.46	0.67	0.44	7.08	8.26	7.86	83.07	0.11	73.74	0.09	86.03	0.12	0.24	7.15
09	56.69	1.01	1.58	14.70	13.01	11.51	89.06	0.13	80.10	0.13	91.92	0.14	1.29	14.88
10	58.47	1.82	0.87	25.63	9.34	27.92	90.86	0.20	82.12	0.18	92.29	0.20	0.46	26.66
11	64.28	2.17	2.60	47.07	14.72	39.07	92.97	0.25	85.59	0.25	94.44	0.27	1.90	50.40
12	68.74	2.83	3.26	77.21	18.58	55.36	89.98	0.39	81.83	0.36	92.37	0.38	2.21	89.28
13	63.79	3.66	3.25	83.64	22.02	54.27	88.43	0.50	83.52	0.44	90.95	0.48	2.80	94.84
14	64.79	6.12	2.16	153.44	17.84	94.49	91.16	0.58	85.64	0.56	92.87	0.57	1.71	171.17
15	70.10	7.35	4.20	195.47	24.44	127.44	91.35	0.81	87.23	0.82	93.87	0.84	3.18	231.52
16	71.55	8.41	5.42	362.04	25.90	187.69	90.84	0.71	87.15	0.74	92.87	0.82	4.94	372.63
17	73.08	11.43	5.24	415.57	26.62	253.60	90.91	0.97	86.98	1.06	92.56	1.07	4.88	470.41
18	69.07	15.26	4.22	516.20	24.91	279.38	92.66	1.26	88.95	1.33	93.84	1.44	3.92	578.00
19	71.70	16.32	4.23	726.58	25.83	397.62	92.51	1.61	89.85	1.63	94.06	1.70	3.79	829.96
20	74.54	23.81	6.34	887.28	26.65	558.24	93.51	1.79	90.08	1.86	94.87	1.99	5.89	1031.59

CPLEX solver was used to solve the linear relaxation of the proposed mathematical formulation applied to instances with up to 20 jobs.

The main contribution of this work is the proposition of five families of valid constraints for SMSPETP formulations based on time-indexed variables. The proposed separation heuristics for these families were also used to obtain lower bounds for instances with up to 20 jobs. The lower bounds obtained with these heuristics are significantly better than those obtained with the linear relaxation of the mathematical formulation presented in this work. Although the times required to generate such lower bounds are greater than those required by CPLEX to solve linear relaxation, the lower bounds obtained are close, or even equal, to the values of the optimal integer solutions.

It is important to note that the valid constraints proposed for the time-indexed SMSPETP formulations can also be used in many other types of scheduling problems involving sequence-dependent setup times.

ACKNOWLEDGEMENTS

The authors thank *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior* - Brazil (CAPES) - Finance Code 001, *Fundação de Amparo à Pesquisa do Estado de Minas Gerais* (FAPEMIG), *Conselho Nacional de Desenvolvimento Científico e Tecnológico* (CNPq), *Centro Federal de Educação Tecnológica de Minas Gerais* (CEFET-MG) and *Universidade Federal de Ouro Preto* (UFOP) for supporting this research.

REFERENCES

[1] Pasquale Avella, Maurizio Boccia, Carlo Mannino, and Igor Vasilyev. 2016. Valid Inequalities for Time-Indexed Formulations of the Runway Scheduling Problem. In *Supplementary Proceedings of the 9th International Conference on Discrete Optimization and Operations Research and Scientific School (DOOR 2016), Vladivostok, Russia, September 19 - 23, 2016.* 787–790. http://ceur-ws.org/Vol-1623/paperapp14.pdf

[2] P. Hansen and N. Mladenović. 2001. Variable neighborhood search: Principles and applications. *European Journal of Operational Research* 130, 3 (2001), 449–467. https://doi.org/10.1016/S0377-2217(00)00100-4

[3] A. Janiak, W. A. Janiak, T. Krysiak, and T. Kwiatkowski. 2015. A survey on scheduling problems with due windows. *European Journal of Operational Research* 242, 2 (2015), 347 – 357. https://doi.org/10.1016/j.ejor.2014.09.043

[4] B. F. Rosa, , M. J. F. Souza, S. R. de Souza, P. Y. P. Michelon, and Z. Ales. 2016. Mathematical formulations for the job scheduling problem with due windows and setup times (in portuguese). In *Proceedings of the XLVIII Brazilian Symposium of Operational Research – XLVIII SBPO.* Vitória, Brazil, 4140–4151.

[5] B. F. Rosa, M. J. F. Souza, S. R. de Souza, M. F. de França Filho, Z. Ales, and P. Y. P. Michelon. 2017. Algorithms for job scheduling problems with distinct time windows and general earliness/tardiness penalties. *Computers & Operations Research* 81 (2017), 203 – 215. https://doi.org/10.1016/j.cor.2016.12.024

[6] J. P. Sousa and L. A. Wolsey. 1992. A time indexed formulation of non-preemptive single machine scheduling problems. *Mathematical Programming* 54, 1 (1992), 353–367. https://doi.org/10.1007/BF01586059

[7] Shunji Tanaka. 2012. An Exact Algorithm for the Single-Machine Earliness-Tardiness Scheduling Problem. In *Just-in-Time Systems*, Roger Z. Ríos-Mercado and Yasmín A. Ríos-Solís (Eds.). Springer Optimization and Its Applications, Vol. 60. Springer New York, 21–40. https://doi.org/10.1007/978-1-4614-1123-9_2

[8] L. A. Wolsey. 1998. *Integer programming.* Wiley-Interscience, New York, NY, USA.