Valid inequalities for a time-indexed formulation☆

Lotte Berghman [a,*], Frits C.R. Spieksma [b]

[a] *Université de Toulouse - Toulouse Business School, 20 BD Lascrosses BP 7010, 31068 Toulouse Cedex 7, France*
[b] *Operations Research Group, KU Leuven, Naamsestraat 69, B-3000 Leuven, Belgium*

ARTICLE INFO

ABSTRACT

We present a general time-indexed formulation that contains scheduling problems with unrelated parallel machines. We derive a class of basic valid inequalities for this formulation, and we show that a subset of these inequalities are facet-defining. We characterize all facet-defining inequalities with right-hand side 1. Further, we show how to efficiently separate these inequalities.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

Consider the following problem. We have a single machine, n jobs (the jobset J), and a discrete timespan (the set of periods T). Each job $j \in J$ has a known processing time that depends on the period $t \in T$ at which job j is started: $p_{j,t}$; there is also a known cost $c_{j,t}$ associated to starting job j at period t. Notice that the period $|T|$ refers to the latest possible period that any job $j \in J$ can start. We assume all data to be integral, and all processing times to be positive. The problem is to schedule all jobs, nonpreemptively, such that the machine needs to work on at most one job in a period, while minimizing total costs.

Here is a formulation, involving binary variables $x_{j,t}$ indicating whether job $j \in J$ starts in period $t \in T$. We use, in this formulation, a set of periods called $A_{j,t}$, which is defined for each $j \in J$, $t \in T$, as follows:

$$A_{j,t} \equiv \{s \leq t : s + p_{j,s} - 1 \geq t\}.$$

Thus, $A_{j,t}$ represents, for a given job j and a given period t, the set of periods s such that if job j starts at period s, job j is still being processed at period t; notice that this set of periods need not be consecutive.

$$\min \quad \sum_{j \in J} \sum_{t \in T} c_{j,t} x_{j,t} \tag{1}$$

such that
$$\sum_{t \in T} x_{j,t} = 1 \quad \forall j \in J, \tag{2}$$

$$\sum_{j \in J} \sum_{s \in A_{j,t}} x_{j,s} \leq 1 \; \forall t \in T, \tag{3}$$

$$x_{j,t} \in \{0, 1\} \quad \forall j \in J, \; \forall t \in T. \tag{4}$$

This formulation is called a time-indexed formulation. One can verify easily that the above formulation is a correct one. We use P to denote the convex hull of feasible solutions to (2)–(4).

Notice that we can generalize this formulation to model a setting where processing a job involves scheduling nonconsecutive periods. Thus, instead of having processing times that consist of a number of consecutive periods, one can imagine that a processing time consists of an arbitrary set of periods that depends on the job j. Then, by viewing the set $A_{j,t}$ as the set of periods s such that if job j is started at period s, the job is being processed at period t, formulation (1)–(4) still applies. Another generalization is to view T as a circular set, without a first, and last, period. In such a setting T could be interpreted as, for instance, a set of positions. Interestingly, many of our results here apply to this setting as well.

1.1. Motivation and literature

There are two main reasons to investigate this formulation. First, model (1)–(4) is quite general. Depending on the choice of cost-coefficients $c_{j,t}$ many well-known scheduling problems arise (see Brucker [3] for a general introduction into scheduling). For instance, in order to minimize total completion time (the completion time of a job is the time at which the execution of the job ends), choose: $c_{j,t} = t + p_{j,t} - 1$; in order to minimize total weighted completion time, choose: $c_{j,t} = w_j(t + p_{j,t} - 1)$; in order to minimize total lateness (the lateness of a job equals

the difference between its completion time and a given due date), choose: $c_{j,t} = t + p_{j,t} - 1 - d_j$; in order to minimize total tardiness (the tardiness of a job is the length of the period with which the due date is violated), choose: $c_{j,t} = \max\{0, t + p_{j,t} - 1 - d_j\}$; in order to minimize total flow time (the flow time of a job is the amount of time elapsed between the given release date of a job and its completion time) $c_{j,t} = t + p_{j,t} - 1 - r_j$. Further, observe that a question like: can all jobs be scheduled before their given deadlines? can be modeled easily by using very large cost-coefficients: choose $c_{j,t} = M$ if $t + p_{j,t} - 1 > d_j$ and $c_{j,t} = 0$ otherwise.

Moreover, although stated as a single machine scheduling problem, in fact, by allowing time-dependent processing times $p_{j,t}$, scheduling problems with multiple machines are contained in this formulation. Let us make this observation more explicit. Consider a set of m machines, a discrete timespan $\{1, 2, \ldots, H\}$, and a set of n jobs to be processed by the machines. For each job j ($1 \le j \le n$), for each machine ℓ ($1 \le \ell \le m$), and for each period t ($1 \le t \le H$), we have a processing time $p_{j,\ell,t}$ and a cost $c_{j,\ell,t}$. The problem is to schedule all jobs by assigning each job to a machine, and to a specific period. We can create an instance of model (1)–(4) as follows. Construct a timespan $T := \{1, 2, \ldots, mH\}$ for a single machine by concatenating the individual timespans of the m different machines. Each job j has to be scheduled on our single machine, and we adapt the processing times and the cost as follows: $p_{j,(\ell-1)H+t} := p_{j,\ell,t}$ and $c_{j,(\ell-1)H+t} := c_{j,\ell,t}$ for each job $j \in J$, $\ell = 1, \ldots, m$, $t = 1, \ldots, H$. To model inequalities (3), we only sum over the periods that belong to the same machine. A solution to the resulting single-machine scheduling problem is easily converted to a solution of the original multiple machine problem. In conclusion, model (1)–(4) can be used to formulate scheduling problems with unrelated parallel machines, and time-dependent processing times.

Second, there are an abundant number of settings sketched in the literature where the processing time of a job depends on its starting time. Apart from specific applications (such as one described in Sawik [11]), the occurrence of more general phenomena as learning effects, and deterioration effects has been well-documented in the literature, see Gawiejnowicz [7]. Learning effects refer to the phenomenon that, over time, a machine (e.g. a worker) becomes better at performing some job. Due to experience, and improved skills, this may lead to faster processing times the later a job is started. Clearly, learning effects, as well as deterioration effects, give rise to time-dependent processing times. Since it is not our intention to give an overview of scheduling problems with learning effects or deterioration effects, we restrict ourselves here to mentioning here the overview papers by Alidaee and Womer [1], Biskup [2], and Cheng et al. [4], and recent work by Qian and Steiner [10] and Kyparisis and Koulamas [8].

Special cases of our model (1)–(4) have been investigated from a polyhedral point of view. In particular, the case where $p_{j,t} = p_j$ for all $j \in J, t \in T$ has received considerable attention starting with the seminal work of Sousa and Wolsey [12], followed by work by Van Den Akker et al. [14,15], and by Crama and Spieksma [5] (who studied the even more special case $p_{j,t} = p$ or all $j \in J, t \in T$). In particular, for the special case where $p_{j,t} = p_j$, [15] describes a column generation procedure for the corresponding time-indexed formulation, and [14] explores the polyhedral properties of the resulting model, characterizing all facet-defining inequalities with right hand side 2. Unlu and Mason [13] perform an experimental evaluation of time-indexed formulations for scheduling problems with multiple machines and stress the relevance of finding valid inequalities for such settings.

1.2. Our results

We present a class of valid inequalities for P. These inequalities generalize inequalities given in Sousa and Wolsey [12]. We show

that a subset of these inequalities are facet-defining, and we prove that, with these inequalities, all facet-defining inequalities with right-hand side 1 are known. Further, we show how to efficiently separate these inequalities.

2. The Generalized Sousa–Wolsey inequalities

We can deduce the following set of valid inequalities. For each $i \in J$, for each $t_2 \in T \setminus \{1\}$, we define the set:

$$R_{i,t_2} = \{t : t \le t_2 - 1 \text{ and } \exists j \in J \setminus \{i\} \text{ such that } t + p_{j,t} - 1 \ge t_2\}.$$

Thus, R_{i,t_2} represents the set of periods $t < t_2$ such that there exists a job $j \ne i$ that, when started at period t, is active at period t_2. Now we introduce, for each $i \in J$, $t_2 \in T \setminus \{1\}$, and for each $t_1 \in R_{i,t_2}$, the following inequality:

$$\sum_{\substack{j \in J \setminus \{i\} \\ s \le t_1}} \sum_{s \in A_{j,t_2}} x_{j,s} + \sum_{s \in A_{i,t_1}} x_{i,s} + \sum_{s=t_1+1}^{t_2} x_{i,s} \le 1. \tag{5}$$

We observe that, in case $p_{j,t} = p_j$ for all j, t, these inequalities are precisely those introduced by Sousa and Wolsey [12]. Therefore, we refer to these inequalities as the Generalized Sousa–Wolsey (or GSW) inequalities. Notice further that by allowing t_1 to be equal to t_2, inequalities (3) arise, allowing us to consider a single class of inequalities containing both (3) and (5). However, for reasons of convenience, we will treat the two classes (3) and (5) separately.

Theorem 1. *The GSW inequalities* (5) *are valid inequalities for* P.

Proof. We say that a variable $x_{j,t}$ is *active* at period s when $t + p_{j,t} - 1 \ge s$. Further, we say that a variable $x_{j,t}$ is *selected* if its value is set to 1. There are three terms in inequalities (5). Observe that each variable from the first term is active at period t_1 as well as at period t_2, each variable from the second term is active at t_1, and each variable from the third term is active at a period t with $t_1 + 1 \le t \le t_2$. Hence, clearly, in any feasible solution, no two variables from any single term can be selected. Also, it follows that selecting a variable from the first term, prohibits selecting any variable from the second or third term. Next, observe that the second term, as well as the third term, only contains variables that correspond to a single job (job i). Thus, no two variables from the second term or third term can both be selected; even more, selecting a single variable from either one of these terms means no variable from the first term can be selected, since any variable from the first term is active at period t_1, at period t_2, and hence also at any period between t_1 and t_2. $\square$

Remark that one can also use integer rounding to derive the GSW inequalities. Indeed, for any $t_2 \ge t_1 + 1$, let $q = t_2 - t_1 + 1$ (thus, q stands for the number of periods between t_1 and t_2 including both t_1 and t_2). Now, select inequalities (3) for each $t = t_1, t_1 + 1, \ldots, t_2$ with fraction $\frac{1}{q}$, and select the equality constraint (2) corresponding to job i with fraction $1 - \frac{1}{q}$. When summing these constraints with their corresponding fractions, and applying integer rounding, we find the GSW inequalities. This shows that the Chvàtal rank of the GSW inequalities is one.

We will now investigate the polyhedral structure of P. More in particular, we will establish the dimension of P, and we will show that the inequalities $x_{j,t} \ge 0$, and a subset of inequalities (3) define facets of P. Also, we will show that a subset of the GSW inequalities define facets of P. These results can be seen as generalizations of results given in Sousa and Wolsey [12], Crama and Spieksma [5], and Van Den Akker et al. [14]. We use techniques similar to the ones used in these papers to derive our results; we point out, however, that not all properties valid for the problem

with $p_{j,t} = p_j$ remain valid for our more general case of time-dependent processing times. For instance, it is no longer true that all inequalities in the linear programming relaxation of (1)–(4) are facet-defining. We refer to Nemhauser and Wolsey [9] for a thorough introduction into polyhedral theory.

We use the following concepts. A *partial feasible* solution is a solution that schedules a subset of the jobs in J in a feasible way (i.e. without overlap). We say that a period t is *free* with respect to a partial feasible solution, when extending the partial feasible solution by scheduling any currently unscheduled job j at period t leads again to a (partial) feasible solution. Further, to avoid technicalities, we will assume that the timespan T is large enough, more concrete, we assume that $|T| \geq (n+6)p_{max}$, where $p_{max} = \max_{j,t} p_{j,t}$.

Theorem 2. *The dimension of the polytope P, $\dim(P) = n(|T| - 1)$.*

Proof. The number of distinct variables $x_{j,t}$ is $n|T|$ as for each job j, there are $|T|$ possible moments to start this job. Since we have n linearly independent equalities (2), it easily follows that $\dim(P) \leq n|T| - n$. We will show that, in fact, equality holds.

Suppose that all feasible solutions x satisfy some equality $\sum_{j \in J} \sum_{t \in T} \alpha_{j,t} x_{j,t} = \beta$. We first show that $\alpha_{j,t} = \alpha_j$ for all $t \in T$. Consider some job $j \in J$ and two distinct periods $s, t \in T$. Schedule all jobs except job j not using the following periods: $\max(s - p_{max} + 1, 1), \ldots, \min(s + p_{max} - 1, |T|)$ and $\max(t - p_{max} + 1, 1) \ldots, \min(t + p_{max} - 1, |T|)$. As a consequence, in the resulting partial feasible solution, the periods s and t are free. Notice that this is always possible: as $|T| \geq (n+6)p_{max}$, enough periods are available to place $n - 1$ jobs.

Starting from this partial feasible solution, construct now a first schedule, obtained by starting job j at period s, and construct a second schedule, obtained by starting job j at period t (while all other jobs remain untouched). Comparing the two schedules, it follows easily that $\alpha_{j,t} = \alpha_{j,s}$ for all $j \in J, s, t \in T$. This implies: $\alpha_j = \alpha_{j,t}$ for all $j \in J, t \in T$. We obtain $\sum_{j \in J} \sum_{t \in T} \alpha_{j,t} x_{j,t} = \sum_{j \in J} \alpha_j \sum_{t \in T} x_{j,t} = \beta$ which is a linear combination of the equalities (2). $\square$

Theorem 3. *For each $j \in J, t \in T$: the inequality $x_{j,t} \geq 0$ defines a facet of P.*

Proof. Let $F = \{x \in P : x_{i,s} = 0\}$ for any $i \in J, s \in T$. Next, assume that $\sum_{j \in J} \sum_{t \in T} \alpha_{j,t} x_{j,t} = \beta$ for all $x \in F$.

The construction used in the proof of Theorem 2 can be easily adapted to show that $\alpha_{j,t} = \alpha_j$ for all $j \in J \setminus \{i\}, t \in T$; we only need to ensure that job i does not start at period s in the partial feasible solution. Further, we use this construction for job i and each pair of distinct periods $s_1, s_2 \in T \setminus \{s\}$, implying that $\alpha_{i,t} = \alpha_i$ for all $t \neq s$.

In conclusion, $\sum_{j \in J} \sum_{t \in T} \alpha_{j,t} x_{j,t} = \sum_{j \in J} \alpha_j \sum_{t \in T} x_{j,t} + \rho x_{i,s} = \beta$ which shows that the equality $\sum_{j \in J} \sum_{t \in T} \alpha_{j,t} x_{j,t} = \beta$ is a linear combination of the inequalities (2) and of $x_{i,s} = 0$. $\square$

Let us now consider inequalities (3). We need the following definition:

$$T^e = \{t \in T : \exists j \in J \text{ with } s + p_{j,s} - 1 = t\}.$$

Thus, $T^e \subseteq T$ represents the set of periods for which it is possible for a job to end. Intuitively, it should be clear that inequalities (3) for which no job ends (that is, for which $t \notin T^e$) are redundant (and hence cannot be facet-defining).

Theorem 4. *For each $t \in T_e$: the inequality $\sum_{j \in J} \sum_{s \in A_{j,t}} x_{j,s} \leq 1$ defines a facet of P.*

Proof. Let $F = \{x \in P : \sum_{j \in J} \sum_{s \in A_{j,t}} x_{j,s} = 1\}$ for any $t \in T_e$ and assume that $\sum_{j \in J} \sum_{t \in T} \alpha_{j,t} x_{j,t} = \beta$ for all $x \in F$.

Consider some job $j \in J$ and two distinct periods $s_1, s_2 \in A_{j,t}$. Schedule all jobs except job j so that periods s_1 and s_2 are free. Construct now a first schedule by starting job j at period s_1 and construct a second schedule by starting job j at period s_2, while all other jobs remain untouched. Notice that both solutions are in F. Comparing both solutions, it follows that $\alpha_{j,s_1} = \alpha_{j,s_2}$ for all $s_1, s_2 \in A_{j,t}$ so we conclude:

$$\alpha_{j,s} = \delta_j \quad \forall j \in J, \ \forall s \in A_{j,t}. \tag{6}$$

Next, consider a job $i \in J$, and a period $s \in T$ such that $s + p_{i,s} - 1 = t$ (since $t \in T_e$ such a job and period must exist). In addition, consider a job $j \in J \setminus \{i\}$ and two distinct periods $s_1, s_2 \notin A_{j,t}$. Schedule all other $n - 2$ jobs so that the periods s and t, as well as the periods s_1 and s_2, are free. Now, in case $s_1 < t$, construct a first schedule by scheduling job j at period s_1, and job i at period t; in case $s_1 > t$, construct a first schedule by scheduling job j at period s_1, and job i at period s. In a similar fashion, construct a second schedule as follows: in case $s_2 < t$, schedule job j at period s_2, and job i at period t; in case $s_2 > t$, construct a second schedule by scheduling job j at period s_2, and job i at period s. Notice that each of the constructed solutions is feasible, and in F. By comparing the first and the second schedule, we find, using (6), for each $i, j \in J, s_1, s_2 \notin A_{j,t}: \alpha_{j,s_1} + \delta_i = \alpha_{j,s_2} + \delta_i$. It follows that $\alpha_{j,s_1} = \alpha_{j,s_2}$ for all $s_1, s_2 \notin A_{j,t}$. We arrive at:

$$\alpha_{j,s} = \alpha_j \quad \forall j \in J, \ \forall s \notin A_{j,t}. \tag{7}$$

Finally, consider a pair of jobs $i, j \in J$ and a period $s \notin (A_{j,t} \cup A_{i,t})$. Schedule all other $n - 2$ jobs so that periods s and t are free. Construct a first schedule by starting job i at period s and job j at period t. Construct a second schedule by starting job j at period s and job i at period t. Again, notice that each solution is feasible, and in F. Comparing both solutions, it follows that $\alpha_{i,s} + \alpha_{j,t} = \alpha_{j,s} + \alpha_{i,t}$ for all $s \notin (A_{i,t} \cup A_{j,t})$. Then, using (6) and (7), we get $\alpha_i + \delta_j = \alpha_j + \delta_i$ for all $i, j \in J$. Equivalently:

$$\delta_j - \alpha_j = \zeta \quad \forall j \in J. \tag{8}$$

Using equalities (6), (7), and (8), we find:

$$\sum_{j \in J} \sum_{t \in T} \alpha_{j,t} x_{j,t} = \sum_{j \in J} \alpha_j \sum_{s \notin A_{j,t}} x_{j,s} + \sum_{j \in J} \delta_j \sum_{s \in A_{j,t}} x_{j,s}$$
$$= \sum_{j \in J} \alpha_j \sum_{t \in T} x_{j,t} + \zeta \sum_{j \in J} \sum_{s \in A_{j,t}} x_{j,s} = \beta,$$

which proves the theorem. $\square$

We now proceed with the GSW inequalities. Similar to inequalities (3), we need a definition to be able to identify those GSW inequalities that define facets of P:

$$R_{i,t_2}^{=} = \{t : t \leq t_2 - 1 \text{ and } \exists j \in J \setminus \{i\} \text{ such that } t + p_{j,t} - 1 = t_2\}.$$

Thus, $R_{i,t_2}^{=} \subseteq R_{i,t_2}$ represents the set of periods $t < t_2$ such that there exists a job $j \neq i$ that, when started at period t, ends at period t_2.

Theorem 5. *For each $i \in J, t_2 \in T \setminus \{1\}, t_1 \in R_{i,t_2}^{=}$: a GSW inequality (5) defines a facet of P.*

Proof. Let us first define the following sets of periods: $B_i = A_{i,t_1} \cup \{t_1 + 1, \ldots, t_2\}$ and, for each $j \in J \setminus \{i\}, C_j = \{s \leq t_1 : s \in A_{j,t_2}\}$. Notice that this allows us to rewrite a GSW-inequality as follows:

$$\sum_{t \in B_i} x_{i,t} + \sum_{j \in J \setminus \{i\}} \sum_{t \in C_j} x_{j,t} \leq 1.$$

Next, let $F = \{x \in P : \sum_{t \in B_i} x_{i,t} + \sum_{j \in J \setminus \{i\}} \sum_{t \in C_j} x_{j,t} = 1\}$ for any $i \in J, t_2 \in T \setminus \{1\}, t_1 \in R_{i,t_2}^{=}$, and assume that $\sum_{j \in J} \sum_{t \in T} \alpha_{j,t} x_{j,t} = \beta$ for all $x \in F$.

Consider two distinct periods $s_1, s_2 \in B_i$. Schedule all jobs except job i so that periods s_1 and s_2 are free. Consider a first schedule, obtained by starting job i at period s_1 and consider a second schedule, obtained by starting job i at period s_2, while all other jobs remain untouched. Comparing both solutions, it follows that $\alpha_{i,s_1} = \alpha_{i,s_2}$ for all $s_1, s_2 \in B_i$. It follows that:

$$\alpha_{i,s} = \delta_i \quad \forall s \in B_i. \tag{9}$$

Next, consider some job $j \in J\setminus\{i\}$, and two distinct periods $s_1, s_2 \in C_j$. Schedule all jobs except job j so that periods s_1 and s_2 are free. Consider a first schedule, obtained by starting job j at period s_1 and consider a second schedule, obtained by starting job j at period s_2, while all other jobs remain untouched. Comparing both solutions, it follows that $\alpha_{j,s_1} = \alpha_{j,s_2}$ for all $s_1, s_2 \in C_j$. It follows that:

$$\alpha_{j,s} = \delta_j \quad \forall j \in J\setminus\{i\}, \forall s \in C_j. \tag{10}$$

Next, consider two distinct periods $s_1, s_2 \notin B_i$, and consider a job $j \in J\setminus\{i\}$ and a period s such that $s + p_{j,s} - 1 = t_2$ (such a job j and period s must exist since $t_1 \in R_{i,t_2}^=$). Schedule all other $n - 2$ jobs so that the periods s and t_1, as well as s_1 and s_2, are free. Construct a first schedule as follows: in case $s_1 < t_1$ start job i at period s_1, and job j at period t_1; in case $s_1 > t_2$ start job i at period s_1, and job j at period s. Next, construct a second schedule as follows: in case $s_2 < t_1$ start job i at period s_2, and job j at period t_1; in case $s_2 > t_2$ start job i at period s_2, and job j at period s, while all other jobs remain untouched. Notice that all solutions are feasible, and in F. Comparing both solutions, it follows that (using (10)): $\alpha_{i,s_1} + \delta_j = \alpha_{i,s_2} + \delta_j$ for all $j \in J\setminus\{i\}$, $s_1, s_2 \notin B_i$. Thus:

$$\alpha_{i,s} = \alpha_i \quad \forall s \notin B_i. \tag{11}$$

In a similar way, we can show that:

$$\alpha_{j,s} = \alpha_j \quad \forall j \in J\setminus\{i\}, \; \forall s \notin C_j. \tag{12}$$

Finally, consider a job $j \in J\setminus\{i\}$ and a period $s \notin (B_i \cup C_j)$. Schedule all jobs, except i and j so that periods s and t_1 are free. Construct a first schedule by starting job i at period s and job j at period t_1. Construct a second schedule by starting job j at period s and job i at period t_1, while all other jobs remain untouched. Notice that both solutions are feasible, and in F. Comparing both solutions, it follows that $\alpha_{i,s} + \alpha_{i,t_1} = \alpha_{j,s} + \alpha_{i,t_1}$ for all $s \notin (B_i \cup C_j)$. Using (9)–(12), it follows that: $\alpha_i + \delta_j = \alpha_j + \delta_i$ for all $j \in J\setminus\{i\}$. Equivalently:

$$\delta_j - \alpha_j = \zeta \quad \forall j \in J. \tag{13}$$

Equalities (9), (10), (11), (12), and (13) together allow us to conclude that:

$$\sum_{j\in J}\sum_{t\in T}\alpha_{j,t}x_{j,t} = \alpha_i\sum_{t\notin B_i}x_{i,t} + \delta_i\sum_{t\in B_i}x_{i,t}$$
$$+ \sum_{j\in J\setminus\{i\}}\alpha_j\sum_{t\notin C_j}x_{j,t} + \sum_{j\in J\setminus\{i\}}\delta_j\sum_{t\in C_j}x_{j,t}$$
$$= \sum_{j\in J}\alpha_j\sum_{t\in T}x_{j,t} + \zeta\left(\sum_{t\in B_i}x_{i,t} + \sum_{j\in J\setminus\{i\}}\sum_{t\in C_j}x_{j,t}\right) = \beta,$$

which proves the theorem. □

A natural question to consider is whether other families of facet-defining inequalities with coefficients in $\{0, 1\}$, and with right-hand side 1 exist. Let us call an inequality where the coefficient of each variable is in $\{0, 1\}$ a *set-packing* inequality. In fact, it turns out that no other facet-defining set-packing inequalities exist for P.

Theorem 6. *All facet-defining inequalities of P that are of the set-packing type and have right-hand side 1, are constraints* (3) *and the GSW-inequalities.*

Proof. Consider some set-packing inequality with right-hand side 1. We will show that any such inequality is implied by an inequality from constraints (3), or by a GSW-inequality (5). In fact, when given some specific set-packing inequality with right-hand side 1, we exhibit either a period t specifying the particular inequality from (3), or we exhibit a job $i \in J$, and two distinct periods $t_1, t_2 \in T$ that define a GSW-inequality. We say that a variable *occurs* in the set-packing inequality when its coefficient equals 1. Next, we say that job $i \in J$ is *disjoint* when the set-packing inequality contains two variables corresponding to the same job, say $x_{i,u}$ and $x_{i,v}$, such that $u + p_{i,u} - 1 < v$. Observe that any set-packing inequality can contain at most one disjoint job (otherwise it is possible to select two jobs that do not overlap, and the inequality is not valid: Imagine two disjoint jobs i and j and four periods u, v, w, z such that $u + p_{i,u} - 1 < v$ and $w + p_{j,w} - 1 < z$. If $u + p_{i,u} - 1 < z$ we can put $x_{i,u} = x_{i,z} = 1$. If on the other hand $w + p_{j,w} - 1 < v$ we can put $x_{j,w} = x_{i,v} = 1$. Remark that at least one of those two conditions always holds. Indeed, if both conditions do not hold, we have $u + p_{i,u} - 1 \geq z$ and $w + p_{j,w} - 1 \geq v$, which, when summed, gives $u + w + p_{i,u} + p_{j,w} - 2 \geq z + v$, contradicting the existence of two disjoint jobs). Thus there are either no disjoint jobs, or there is one disjoint job.

In case there are no disjoint jobs, find, in the set-packing inequality, the largest period for which there is a variable occurring in the inequality with coefficient 1: call this period s. Due to the absence of disjoint jobs, it follows that any variable occurring in the set-packing inequality must be active at period s. This implies that an inequality (3) with $t = s$ implies this set-packing inequality.

Next, let us assume that there is one disjoint job. Let the disjoint job be job i. Further, let v be the latest period for which variable $x_{i,v}$ occurs in the set-packing inequality, and let u be the period for which $u + p_{i,u}$ is minimal over all variables $x_{i,t}$ occurring in the set-packing inequality. We claim that a GSW inequality with job i, and with:

$$t_1 = u + p_{i,u} - 1$$
$$t_2 = v,$$

implies the set-packing inequality.

Indeed, the set-packing inequality cannot contain variables $x_{j,t}$ with $j \neq i$, and $t \geq t_1 + 1$ (otherwise this variable can be selected together with $x_{i,u}$). Also, the set-packing inequality can only contain variables $x_{j,t}$ with $j \neq i$, in case such a variable is active at period t_2 (otherwise this variable can be selected together with $x_{i,v}$). Finally, notice that by definition of u and v, and by the implied corresponding values of t_1 and t_2, the set-packing inequality does not contain variables $x_{i,t}$ such that $t \geq t_2 + 1$, and, with respect to variables $x_{i,t}$ with $t \leq t_1$, each such variable occurring in the set-packing inequality is active at period t_1. □

3. Separation of GSW inequalities

In this section, we consider the computational task of separating the GSW inequalities (5). Clearly, when solving instances of a time-indexed formulation using a cutting-plane algorithm, being able to efficiently separate the GSW-inequalities is of interest.

The separation problem corresponding to the GSW inequalities can be stated as follows:

INPUT: a vector x satisfying (2), (3), and $x \geq 0$.
QUESTION: does there exist a GSW inequality violated by x?

Recall that each GSW inequality is defined by specifying some job $i \in J$, and two distinct periods $t_1, t_2 \in T$. A naive search for a violated GSW inequality would consider each possible choice for a job $i \in J$, and each possible pair of distinct periods $t_1, t_2 \in T$. Next, by determining for each variable $x_{j,t}$ whether it is contained in the specific GSW inequality defined by $i \in J, t_1, t_2 \in T$, and, in this manner, computing the left hand side of each inequality in (5), leads to an $O(n|T|^2 \times n|T|) = O(n^2|T|^3)$ complexity.

However, a more efficient method is possible. First, let us take the following point of view (see Dokka et al. [6]): the vector x can be described by the *support* of x, that is by a list of those $x_{j,t}$ variables that have a positive value. This is relevant since the number of variables of the x-vector equals $n \times |T|$, whereas the number of positive entries in the vector x can be much smaller. Indeed, observe that any basic feasible solution to the linear programming relaxation of (1)–(4) can be described by $|T|$ basic variables. All this has the effect of making the description of the input to the separation problem more compact. Thus, given some x, we view the *support* of x as a set S of all pairs of indices (j, t) such that $x_{j,t} > 0$, and we see $|S|$ as a measure of the size of the input.

Second, there is structure present in any violated GSW inequality as described in the next lemma.

Lemma 7. *If x violates a GSW inequality and satisfies inequalities* (3), *then there exists a violated GSW inequality defined by parameters $i \in J, t_1, t_2 \in T$ such that the support of x contains*

(i) x_{i,t_2}, *and*
((ii) $x_{i,t}$ *for some* $t \in A_{i,t_1}$, *or*
(iii) x_{j,t_1} *for some* $j \in J \backslash \{i\}$).

Proof. We argue by contradiction. First, we deal with (i). Suppose x violates a specific GSW inequality, determined by parameters i, t_1, t_2, with $x_{i,t_2} = 0$. There are two cases: either $t_1 = t_2 - 1$ or $t_1 \leq t_2 - 2$. In the first case, since $x_{i,t_2} = 0$, it follows that each variable in the violated GSW inequality is active at period t_1, and hence this violated GSW inequality is implied by an inequality from (3) with $t = t_1$. This, however is impossible by the assumption in the lemma, and hence we are in the second case $t_1 \leq t_2 - 2$. Then, we claim that the GSW-inequality determined by parameters $i, t_1, t_2 - 1$ (called the 'new' inequality) also corresponds to a violated GSW-inequality. This follows from the observation that the only variable that appears in the former GSW-inequality, and not in the 'new' GSW inequality is x_{i,t_2} (which equals 0).

Consider now (ii). Suppose x violates a specific GSW inequality, determined by parameters i, t_1, t_2, with $x_{i,t} = 0$ for all $t \in A_{i,t_1}$. Again, there are two cases: either $t_1 = t_2 - 1$ or $t_1 \leq t_2 - 2$. In the first case, since $x_{i,t} = 0$ for all $t \in A_{i,t_1}$, it follows that each variable in the violated GSW inequality is active at period t_2, and hence this violated GSW inequality is implied by an inequality from (3) with $t = t_2$. This, however is impossible by the assumption in the lemma, and hence we are in the second case $t_1 \leq t_2 - 2$. Then, we claim that the GSW-inequality determined by parameters $i, t_1 + 1, t_2$ also corresponds to a violated GSW-inequality. This follows from the observation that the only variables that are in the former GSW-inequality, and not in the new GSW inequality are $x_{i,t}$ with $t \in A_{i,t_1} \backslash A_{i,t_1+1}$ (which are all equal to 0).

Finally, we deal with (iii). Suppose x violates a specific GSW inequality, determined by parameters i, t_1, t_2, with $x_{j,t_1} = 0$ for all $j \in J \backslash i$. Then, we claim that the GSW-inequality determined by pa-

rameters $i, t_1 - 1, t_2$ also corresponds to a violated GSW-inequality. This follows from the observation that the only variables that are in the former GSW-inequality, and not in the 'new' GSW inequality are x_{j,t_1} for all $j \in J \backslash \{i\}$ (which are all equal to 0). $\square$

Using Lemma 7, and in particular properties (i) and (iii), we specify the following algorithm to establish whether, given some x, a violated GSW inequality exists. As mentioned before, S denotes the support of x.

Algorithm 1 Separation algorithm for GSW inequalities
---
  for all $(j, t) \in S$ **do**
    set $i := j, t_2 := t$
    for all $(j, t) \in S$ with $j \neq i, t \leq t_2 - 1$ **do**
      $t_1 := t$
    end for
  end for
  Test whether a GSW inequality defined by i, t_1, t_2 is violated

---

Theorem 8. *GSW inequalities can be separated in $O(|S|^3)$.*

Proof. Lemma 7 implies that we can restrict our search for a violated GSW inequality, i.e., to a job $i \in J$, and two distinct periods $t_1, t_2 \in T$, for which $(i, t_2) \in S$ and $(j, t_1) \in S$ for some $j \neq i$. This ensures that Algorithm 1 finds a violated GSW inequality whenever there exists one. Its complexity is determined by the two "for" loops, and by determining, for each element in the set S, whether it corresponds to a variable occurring in the specific GSW inequality: $O(|S|^3)$. $\square$

References

[1] B. Alidaee, N.K. Womer, Scheduling with time dependent processing processing times: review and extensions, J. Oper. Res. Soc. 50 (1999) 711–720.
[2] D. Biskup, A state-of-the-art review on scheduling with learning effects, European J. Oper. Res. 188 (2008) 315–329.
[3] P. Brucker, Scheduling Algorithms, Springer, Berlin, 2007.
[4] T.C.E. Cheng, Q. Ding, B.M.T. Lin, A concise survey of scheduling with time-dependent processing times, European J. Oper. Res. 152 (2004) 1–13.
[5] Y. Crama, F.C.R. Spieksma, Scheduling jobs of equal length: complexity, facets and computational results, Math. Program. 54 (1996) 353–367.
[6] T. Dokka, I. Mourtos, F.C.R. Spieksma, Fast separation algorithms for multi-dimensional assignment problems, in: Proceedings of the 2nd International Symposium on Combinatorial Optimization, in: Lecture Notes in Computer Science, vol. 7422, 2011, pp. 189–200.
[7] S. Gawiejnowicz, Time-Dependent Scheduling, Springer-Verlag, Berlin, Heidelberg, 2008.
[8] G.J. Kyparisis, C. Koulamas, On index priority sequencing rules for scheduling with time-dependent job processing times, Oper. Res. Lett. 41 (2013) 445–448.
[9] G. Nemhauser, L.A. Wolsey, Integer and Combinatorial Optimization, Wiley, New York, 1988.
[10] J. Qian, G. Steiner, Fast algorithms for scheduling with learning effects and time-dependent processing times on a single machine, European J. Oper. Res. 188 (2013) 315–329.
[11] T. Sawik, An integer programming approach to scheduling in a contaminated area, Omega 38 (1992) 179–191.
[12] J.P. Sousa, L.A. Wolsey, A time-indexed formulation of non-preemptive single-machine scheduling problems, Math. Program. 72 (1992) 207–227.
[13] Y. Unlu, S.J. Mason, Evaluation of mixed integer programming formulations for nonpreemptive parallel machine scheduling problems, Comput. Ind. Eng. 58 (2010) 785–800.
[14] J.M. Van Den Akker, C.A.J. Hurkens, M.W.P. Savelsbergh, Time-indexed formulations for machine scheduling problems: column generation, INFORMS J. Comput. 12 (2000) 111–124.
[15] J.M. Van Den Akker, C.P.M. Van Hoesel, M.W.P. Savelsbergh, A polyhedral approach to single-machine scheduling problems, Math. Program. 85 (1999) 541–572.