Scheduling Two Chains of Unit Jobs on One Machine: A Polyhedral Study

Claudio Arbib * Martine Labbé † Mara Servilio ‡

January 14, 2011

Abstract

We investigate polyhedral properties of the following scheduling problem: given two sets of unit, indivisible jobs and revenue functions of the jobs completion times, find a one-machine schedule maximizing the total revenue under the constraint that the schedule of each job set respects a prescribed chain-like precedence relation. A solution to this problem is an order preserving assignment of the jobs to a set of time-slots. We study the convex hull of the feasible assignments, and provide families of facet-defining inequalities in two cases: (i) each job must be assigned to a time-slot, (ii) a job does not need to be assigned to any time-slot.

Keywords: One-Machine Scheduling, Convex Hull, Facet-defining Inequalities.

1 Introduction

This paper investigates polyhedral aspects of a one-machine scheduling problem where two sets of unit jobs compete for a single resource over time. Specifically, the problem is:

Problem 1 Given unit indivisible jobs $j \in J = J_1 \cup J_2$, a discrete planning horizon $T$ and, for each $j$, a utility function $f_j : T \to \mathbb{R}$, assign a completion

---

*Università degli Studi di L’Aquila, Dipartimento di Informatica, via Vetoio, I-67010 Coppito, L’Aquila, Italy; e-mail: claudio.arbib@univaq.it
†Université Libre de Bruxelles, Département d’Informatique, Boulevard du Triomphe, B-1050 Brussels, Belgium; e-mail: mlabbe@ulb.ac.be
‡Università degli Studi di L’Aquila, Dipartimento di Informatica, via Vetoio, I-67010 Coppito, L’Aquila, Italy; e-mail: mara.servilio@univaq.it
time \( C_j \in T \) to each \( j \in J \) so that \( \sum_{j \in J} f_j(C_j) \) is maximized, under the constraint that the jobs of \( J_k \) \((k = 1, 2)\) must be processed in a strict order given a priori.

As customary in scheduling problems, the prescription \( i \prec j \) \((i \text{ precedes } j)\) requires that job \( j \) can be scheduled only if job \( i \) has been completed. In our case, such a prescription holds independently for the jobs in \( J_1 \) and \( J_2 \), under the form of two chain-like precedence relations. We here address both the \textit{complete} and the \textit{incomplete} scheduling problem, that is, the case in which all the jobs must be scheduled and the case in which some (last in the chain) may not. Following the usual three-field notation \cite{7} the problem is denoted by \( 1|p_j = 1, 2 \text{ chains}|\sum f_j(C_j) \). Both the complete and the incomplete problems can be solved in polynomial time by dynamic programming.

\textit{Related problems and applications}

A closely related problem with the same set of feasible solutions is studied in Arbib et al. \cite{3}: two users of a telecommunication system hold different sets of unit jobs \( J_k \) which must be assigned to a single channel. The assignment must respect appropriate requirements, reducing to chain-like precedence relations between the jobs belonging to the same set \( J_k \). The objective is the maximization of the least utility gained by any user, where the utility of a user is the sum of the individual revenues \( f_j(C_j) \) of the jobs of the relevant set. Unlike the one considered in this paper, such a problem is NP-hard, even for regular utility functions, i.e., \( f_j \) non-increasing with the completion time of \( j \).

A similar problem is the \textit{Sequence Alignment Problem} arising in computational biology and described by Lancia et al. \cite{8}. This problem calls for comparing two biological entities represented as linearly ordered sets with the aim of discovering the most likely order-preserving correspondence, where the likelihood of a correspondence is the sum of individual weights, one for each corresponding pair. The problem is equivalent to finding a one-machine schedule of two sets of unit jobs obeying a relaxed chain-like precedence constraint: job \( i \) cannot be scheduled after job \( j \) whenever \( i \prec j \). Such a constraint is a relaxation of a chain-like precedence relation, because if job \( j \) is scheduled, then either job \( i \) is scheduled before it, or it is not scheduled at all.

Another interesting application of job-to-slot assignment with chain-like precedence relation arises in air traffic flow management: the same aircraft is employed in consecutive flights whose departures must be assigned to time-slots, and departures can be delayed – at a certain cost – according to aircraft availability. A model with \( m \) users (aircrafts) arranged in \( p \)-chains gives a \( p \)-machine \( m \)-job flowshop. Rossi and Smriglio \cite{18} formulated the case \( p = 2 \) as
a stable set problem.

Literature

Single machine non-preemptive scheduling problems have been the subject of earlier polyhedral studies.

Queyranne and Schulz [16] consider an integer programming formulation where the decision variables are the jobs completion times $C_j$. Although this formulation includes general precedence constraints and processing times, it only encompasses linear objective functions of the form $\sum w_j C_j$.

In van den Akker et al. [20] a time-indexed 0-1 linear programming formulation is proposed for a problem with general processing times, but no precedence constraints. The advantage of a time-indexed formulation (where solutions are assignments of $J$ to $T$) is that any separable objective function of the form $\sum f_j(C_j)$ can be expressed as a linear function of the time-indexed variables. In [20], schedules are initially constrained to be complete (i.e., every job must be scheduled at some time-slot), but since the resulting polyhedron is not full-dimensional, the study focuses on a polyhedron where this constraint is relaxed.

Crama and Spieksma [5] studied the convex hull of the set of feasible complete schedules for problems in which the jobs have equal processing times. Further contributions on time-indexed formulations can be found in Waterer et al. [21].

Starting from a time-indexed formulation, Möhring et al. [11, 12] formulate a general project scheduling problem with time lags as a minimum cut problem on a directed acyclic graph. Scheduling on one machine all the jobs of a single chain is a particular case of this problem (instead, the incomplete problem cannot be reformulated in such terms for general $f_j(C_j)$).

Although polyhedral results cannot be directly drawn from the above reformulation, a full description of the polyhedron of the single chain complete problem was already given by Chaudhuri et al. [6]. The incomplete problem in which intermediate jobs of the chain may not be scheduled was studied by Alevras [2], who provided a complete description of the relevant polytope.

Motivation

Polyhedral studies of polynomial-time solvable problems are pretty frequent, since a complete or partial description of the convex hull of feasible solutions can be a valid support for solving NP-hard problems arising when adding further constraints.

A well-known case is the Stable Set problem on claw-free graphs: although it can be solved in polynomial time, [9], obtaining a complete description of the convex hull is a long-standing open problem, see e.g. Pecher and Wagler [13].
Another important stream of studies addresses the Lot Sizing problem, that in many cases admits a polynomial dynamic programming algorithm. According to A. Atamtürk and K. Küçükyavuz [4], “it has been demonstrated in earlier studies [14, 23] that a good understanding of the polyhedral structure of single-item lot-sizing problems can be very useful in solving more complicated problems, involving multiple products and stages”.

For an example closer to ours, Queyranne and Wang [17] investigate the convex hull of the feasible schedules of a non-preemptive single-machine scheduling problem with precedence constraints. The result is (i) a complete description of the minimal linear system defining the scheduling polyhedron under specific precedence constraints, called *series-parallel*, that make the problem polynomial, and (ii) a partial description of the polyhedron when general precedence constraints are considered.

In our case, it is easy to see that the addition of arbitrary incompatibility constraints between job-slot pairs makes the problem NP-hard. A typical situation is when the feasibility of an assignment depends on the availability of a special resource (e.g., manpower) that obeys independent constraints or schedules. In this case dynamic programming cannot be applied any longer; on the other hand, it is not difficult to find examples in which the bound provided by the straightforward time-indexed formulation (see Section 2) is definitely improved by valid inequalities defining facets of the polyhedron of the unconstrained polynomial problem.
This paper

Our contribution is a polyhedral study that shares some aspects with [14, 17, 23], because Problem 1 admits an exact algorithm that runs in polynomial time, and with [20], because the result is achieved by a time-indexed formulation.

We provide facet-defining inequalities for both the incomplete and the complete problem.

Part of the results for the incomplete problem derive from a reformulation as a stable set in the associated conflict graph, whereas other facet-defining inequalities are problem-specific and cannot be inherited in this way. From these results, facets for \( m = 1 \) chain are easily obtained: we prove that those facets fully describe the polyhedron.

Facet-defining inequalities for the complete problem are obtained via the projection technique introduced by Wolsey [22] and later extended by Martin et al. [10].

The paper is organized as follows. In Section 2 we give a time-indexed formulation of the general problem with \( m \) chains, discuss some known polyhedral properties and introduce the problem variants. The remainder of the paper focuses on the case of \( m = 2 \) chains: polyhedral studies for the incomplete and complete problem are respectively presented in Sections 3 and 4.

2 Problem formulation, general properties and known results

In this section we formulate using integer linear programming the general problem of scheduling \( m \) chains and point out some of its properties.

We represent jobs and time-slots by consecutive indices:

\[
J_k = \{1, 2, \ldots, n_k\}, \quad k = 1, \ldots, m \quad \text{and} \quad T = \{1, 2, \ldots, n\}
\]

assuming \( n \geq n_1 + \cdots + n_m \) and \( i \prec j \) if and only if \( i < j \), for any \( i, j \in J_k \).

Since the same index can denote jobs in different \( J_k \), a superscript \( k \) will be used when necessary.

Define 0-1 variables \( x_{jt}^k \) for any job \( j = 1, \ldots, n_k \) \((k = 1, \ldots, m)\) and for any slot \( t = j, \ldots, n \). Each variable specifies the assignment of a job to a slot, precisely, \( x_{jt}^k = 1 \) if and only if \( j \) is scheduled at \( t \). Observe that \( x_{jt}^k \) is not defined for \( t < j \), since \( j \) can be scheduled only if all the \( j - 1 \) jobs preceding it are scheduled before it. Hence, an assignment is represented by a vector \( \mathbf{x} = (\mathbf{x}^1, \ldots, \mathbf{x}^m) \) where \( \mathbf{x}^k, k = 1, \ldots, m, \) is a subvector of dimension
\begin{align*}
(n + 1)n_k - \frac{n_k(n_k + 1)}{2}: \text{ let } d = (n + 1) \sum_{k=1}^{m} n_k - \sum_{k=1}^{m} \frac{n_k(n_k + 1)}{2} \text{ denote the dimension of } x.
\end{align*}

Let \( f^k_j(t) \) be the revenue obtained by scheduling, and therefore completing, \( j \in J_k \) at time \( t \). For any \( k = 1, \ldots, m \) the problem can be written:

\begin{equation}
\max \sum_{k=1}^{m} \sum_{j=1}^{n_k} f^k_j(t) x^k_{jt}
\end{equation}

\begin{align}
\sum_{t=j}^{n} x^k_{jt} & \leq 1 \quad 1 \leq k \leq m, 1 \leq j \leq n_k \quad (2) \\
\sum_{k=1}^{m} \sum_{j=1}^{\min(t,n_k)} x^k_{jt} & \leq 1 \quad 1 \leq t \leq n \quad (3) \\
x^k_{jt} - \sum_{s=j-1}^{t-1} x^k_{j-1,s} & \leq 0 \quad 1 \leq k \leq m, \quad j \geq 2, \quad t \geq j \quad (4) \\
x^k_{jt} & \geq 0 \quad 1 \leq j \leq n_k, \quad j \leq t \leq n \quad (5) \\
x^k_{jt} & \text{ integer} \quad 1 \leq j \leq n_k, \quad j \leq t \leq n \quad (6)
\end{align}

The meaning of inequalities (2), (3) is evident. Notice that a job (respectively, a slot) does not need to be assigned to a slot (respectively, a job): in fact, (1)-(6) formulate a variant called the \textsc{Partial Scheduling Problem}, \textsc{PSP}_m (the index refers to the number of chains).

Inequalities (4) express the precedence relations among the jobs of \( J_k \): if \( j \) is not the first job of \( J_k \) and is scheduled at \( t \), then \( j - 1 \) must be scheduled at some \( s < t \). An enforcement of such inequalities, found in [11], holds for all \( 1 \leq k \leq m, j \geq 2 \) and \( t \geq j \):

\begin{equation}
\sum_{s=j}^{t} x^k_{js} - \sum_{s=j-1}^{t-1} x^k_{j-1,s} \leq 0 
\end{equation}

Inequalities (7) can be obtained from (4) by lifting variables in any order. The optimal lifting coefficients of variables in \( L = \{ x^k_{js} : s = j, \ldots, t-1 \} \) are equal to 1; the remaining ones are 0.

The formulation simplifies after the observations below.

\textbf{Observation 2.1} Inequalities \( x^k_{jj} \geq 0 \) are redundant for \( j < n_k \).

\textit{Proof.} Inequality (7) for \( j = t = n_k \) gives

\begin{equation*}
x^k_{n_k-1,n_k-1} \geq x^k_{n_k,n_k}.
\end{equation*}

The right-hand side being nonnegative, we can conclude that \( x^k_{n_k-1,n_k-1} \geq 0 \), and so on. \( \square \)
Observation 2.2 Inequalities (2) are redundant for \( j \geq 2 \).

Proof. Inequalities (2) (for \( j = 1 \)), (7) (for \( j = 2 \)) and (5) imply

\[
\sum_{s=2}^{n} x_{s}^{k} \leq \sum_{s=1}^{n-1} x_{s}^{k} \leq \sum_{s=1}^{n} x_{s}^{k} \leq 1
\]

i.e., inequality (2) for \( j = 2 \), and so on. \( \Box \)

To require that all jobs are scheduled, one has to suppress variables \( x_{jt}^{k} \) for \( t > n - n_{k} + j \) and replace constraints (2) by

\[
\sum_{t=j}^{n} x_{jt}^{k} = 1 \quad 1 \leq j \leq n_{k}
\]

In this case, Observation 2.2 is not valid anymore. We refer to this variant as the COMPLETE SCHEDULING PROBLEM, CSP\(_{m}\). A further variant of complete scheduling occurs when \( m > 1 \) and \( n = \sum n_{k} \). For such a variant, called the SHUFFLING PROBLEM, ShP\(_{m}\), Observation 2.2 is replaced by

Observation 2.3 Inequalities (8) are redundant for \( j < n_{k} \).

Proof. Inequality (8) for \( j = n_{k} \) and (7) for \( j = n_{k} - 1 \) imply

\[
\sum_{s=n_{k}-1}^{n-1} x_{s,n_{k}-1}^{k} \geq \sum_{s=n_{k}}^{n} x_{s}^{k} = 1
\]

The leftmost term is prevented from being > 1 by \( \sum_{k=1}^{m} n_{k} = n \). The argument can be repeated for \( j = n_{k} - 1 \), and so on. \( \Box \)

We denote the convex hulls of feasible solutions of PSP\(_{m}\), CSP\(_{m}\) as

\[
P_{PS}^{m} = \text{conv}(\{ x \in \mathbb{R}^{d} : (2)-(3),(5)-(7) \}),
\]

\[
P_{CS}^{m} = \text{conv}(\{ x \in \mathbb{R}^{d} : (3),(5),(6)-(8) \}).
\]

(ShP\(_{m}\) can be treated as a particular case of CSP\(_{m}\), as shown in Section 4.2 for \( m = 2 \).)

The following known result holds for PSP\(_{m}\).

Theorem 2.1 \( P_{PS}^{m} \) is full-dimensional.

It is also easy to see that
Theorem 2.2  *Inequalities*

\[
\sum_{t=1}^{n} x_{1t}^k \leq 1 \quad (9)
\]

\[
x_{jt}^k \geq 0 \quad t > j \quad (10)
\]

\[
x_{nk_{nk}}^k \geq 0
\]

induce facets of \( P^m_{PS} \) for \( k = 1, \ldots, m \).

In the following two sections we address \( m = 2 \), i.e., the polyhedra \( P^2_{PS} \) (Section 3), \( P^2_{CS} \) and \( P^2_{Sh} \) (Section 4).

### 3  PSP$_2$

Let us simplify notation as follows

- \( x_{jt}^1 = x_{jt}, f_1^j(t) = a_{jt} \) for all \( 1 \leq j \leq n_1 \) and \( t \geq j \)
- \( x_{jt}^2 = y_{jt}, f_2^j(t) = b_{jt} \) for all \( 1 \leq j \leq n_2 \) and \( t \geq j \)

A solution is then a pair \((x, y)\) of subvectors representing the assignments of \( J_1 \) and \( J_2 \), respectively. Also, \( e_k^t \), for \( k = 1 \) or \( 2 \), denotes the unit vector assigning the unique job \( j \in J_k \) to slot \( t \), i.e. \( e_1^t = 1 \) and \( e_2^t = 0 \) for all \((i, s) \neq (j, t)\). In other words, \( x = e_k^t \) assigns \( 1 \leq j \leq n_k \) to \( t \), and no other \( j' \in J_1 \cup J_2 - \{j\} \) to
any slot. Formulation (1)-(6) can be rewritten as

\[
\text{max} \sum_{i=1}^{n_1} \sum_{t=1}^{n} a_{it} x_{it} + \sum_{j=1}^{n_2} \sum_{t=j}^{n} b_{jt} y_{jt} \tag{11}
\]

\[
\begin{align*}
\sum_{t=1}^{n} x_{1t} & \leq 1 \\
\sum_{t=1}^{n} y_{1t} & \leq 1
\end{align*}
\]

\[
\begin{align*}
\sum_{i=1}^{\min\{t,n_1\}} x_{it} + \sum_{j=1}^{\min\{t,n_2\}} y_{jt} & \leq 1 & 1 \leq t \leq n \\
\sum_{s=i}^{t} x_{is} - \sum_{s=i-1}^{t-1} x_{i-1,s} & \leq 0 & 2 \leq i \leq n_1, \ t \geq i \\
\sum_{s=j}^{t} y_{js} - \sum_{s=j-1}^{t-1} y_{j-1,s} & \leq 0 & 2 \leq j \leq n_2, \ t \geq j
\end{align*}
\]

Theorem 2.2 ensures that the first and second set of constraints, and the non-negativity constraints, are facet-inducing. Let us prove the following

**Theorem 3.1 Inequalities**

\[
\begin{align*}
\sum_{s=i}^{t} x_{is} - \sum_{s=i-1}^{t-1} x_{i-1,s} & \leq 0 & 2 \leq i \leq n_1, \ t \geq i \\
\sum_{s=j}^{t} y_{js} - \sum_{s=j-1}^{t-1} y_{j-1,s} & \leq 0 & 2 \leq j \leq n_2, \ t \geq j
\end{align*}
\]

induce facets of $P^2_{P_S}$.

**Proof.** Let $i \in J_1, t \in T$, and let $H = \{x \in \mathbb{R}^{n_1n} : \pi x = \pi_0\}$ denote a hyperplane of $\mathbb{R}^{n_1n}$. Suppose that $H$ contains 0; hence, $\pi_0 = 0$. For all $1 \leq s \leq n$, let $x^{is} = e^{is}$. Define then

\[
x^{is} = \sum_{h=1}^{j-1} e^{hs} + e^{js}
\]
for all $1 \leq j \leq n_1$ and $s \geq j$ such that (a) $j \neq i - 1, i$ and $s \geq j$, or (b) $j = i - 1$ and $s \geq t$, or (c) $j = i$ and $i \leq s \leq t$. Suppose $x^j s \in H$. For $j \leq i - 2$, we get

$$\pi_{j, s} = 0 \quad \text{for } 1 \leq j \leq i - 2, s \geq j.$$ 

Solving $\pi x^{i - 1, s} = 0$ for $s \geq t$, we get

$$\pi_{i - 1, s} = 0 \quad \text{for } s \geq t,$$

and solving $\pi x^i s = 0$ for $i \leq s \leq t$ produces

$$\pi_{i, s} = -\pi_{i - 1, i - 1} \quad \text{for } i \leq s \leq t.$$

Take now

$$u^{i - 1, s} = \sum_{h=1}^{i - 2} e^{hh} + e^{i - 1, s} + e^{it} \quad \text{for } i - 1 \leq s < t$$

$$w^i s = \sum_{h=1}^{i - 2} e^{hh} + e^{i - 1, t} + e^{is} \quad s > t$$

Solving then $\pi u^{i - 1, s} = 0$, $\pi w^i s = 0$, we now obtain

$$\pi_{i - 1, s} = -\pi_{i t} \quad \text{for } i - 1 \leq s < t$$

$$\pi_{i, s} = 0 \quad \text{for } s > t.$$ 

Finally, taking $x^j s$ for $j > i, s \geq j$ and solving again $\pi x^j s = 0$, we see that the remaining components of $\pi$ equal 0. Thus, we have found $n_1 n$ affinely independent points in $H = \{x \in P_{PS} : \sum_{s=1}^{t} x_{is} = \sum_{s=1}^{t-1} x_{i-1, s}\}$.

The result for inequalities (13) is proved by a similar procedure. \(\square\)

None of the facet-inducing inequalities given so far involve variables associated with both $J_1$ and $J_2$. The remaining inequalities presented in this section can be derived using a conflict graph $G$ with vertices corresponding to variables $x_{it}$ and $y_{jt}$. Some of these inequalities can be classified according to special subgraphs of $G$; for others, a special structure is identified.

### 3.1 Clique inequalities

The four families of inequalities dealt with in the following Theorems 3.2-3.5 are associated with maximal cliques of $G$. All these inequalities are of the form

$$x(S_1) + y(S_2) \leq 1,$$
where $S_k \subseteq J_k \times T$: that is, in a feasible schedule no more than one variable of $S_1 \cup S_2$ can be set to 1.

**Theorem 3.2 Inequalities**

\[
\min_{t,n_1} \sum_{i=1}^{\min\{t,n_1\}} x_{it} + \min_{t,n_2} \sum_{j=1}^{\min\{t,n_2\}} y_{jt} \leq 1 \quad 1 \leq t \leq n
\] (14)

induce facets of $P_{PS}^2$.

**Proof.** For $t \in T$, let

\[
F_t = \{z = (x, y) \in P_{PS}^2 : \sum_{i=1}^{\min\{t,n_1\}} x_{it} + \sum_{j=1}^{\min\{t,n_2\}} y_{jt} = 1\}
\]

and let $\pi z = \pi_0$ be any hyperplane containing $F_t$. Let us show that

\[
\pi z = \pi_0[\sum_{i=1}^{\min\{t,n_1\}} x_{it} + \sum_{j=1}^{\min\{t,n_2\}} y_{jt}].
\]

Define

\[
p^{1t} = (e^{1,1,t}, 0), \quad q^{1t} = (0, e^{2,1,t}).
\]

Since $p^{1t}, q^{1t} \in F_t$, $\pi p^{1t} = \pi q^{1t} = \pi_0$. One then obtains

\[
\pi^{1,1,t} = \pi^{2,1,t} = \pi_0, \quad \text{for all } t \in T.
\]

For $i = 1, \ldots, \min\{t,n_1\}, r \geq i, r \neq t$ and $j = 1, \ldots, \min\{t,n_2\}, s \geq j, s \neq t$, define

\[
w^{ir} = (\sum_{h=1}^{i-1} e^{1,h,h} + e^{1,i,r}, e^{1,2,t}),
\]

\[
z^{js} = (e^{1,1,t}, \sum_{h=1}^{j-1} e^{2,h,h} + e^{2,j,s}).
\]

Again, replacing $w^{ir}, z^{js}$ in $\pi z = \pi_0$ and solving, one gets

\[
\pi^{1,i,r} = \pi^{2,j,s} = 0.
\]
For $i = 2, \ldots, \min\{t, n_1\}$ and $j = 2, \ldots, \min\{t, n_2\}$, take now

\[
\begin{align*}
u^j_t &= (0, \sum_{h=1}^{j-1} e^{2.h,h} + e^{2.j,t}), \\
u^{j,t} &= (0, \sum_{h=1}^{j-1} e^{2.h,h} + e^{2.j,t}).
\end{align*}
\]

Then

\[\pi^{1,i,t} = \pi^{2,j,t} = \pi_0.\]

Finally, if $\min\{t, n_1\} = t$ solve $\pi w^{ir} = \pi_0$ (if $\min\{t, n_2\} = t$ solve $\pi z^{is} = \pi_0$) for all $i > t$ and $r \geq i$ ($j > t$ and $s \geq j$), getting

\[\pi^{1,i,r} = 0 \quad \text{and} \quad \pi^{2,j,s} = 0.\]

The same technique as that employed in Theorem 3.2 can be used to show that all the inequalities below are facet-inducing for $P^2_{PS}$ (Theorems 3.3-3.5). Proofs are omitted for the sake of shortness: the reader is referred to Servilio [19] for details.

**Theorem 3.3 Inequalities**

\[
\sum_{s=1}^{t} x_{is} + \sum_{s=t-i+1}^{t} y_{t-i+1,s} \leq 1 \tag{15}
\]

for $2 \leq i \leq n_1$ and $t \geq i$, and

\[
\sum_{s=j}^{t} y_{js} + \sum_{s=t-j+1}^{t} x_{t-j+1,s} \leq 1 \tag{16}
\]

for $2 \leq j \leq n_2$ and $t \geq j$, are valid and induce facets of $P^2_{PS}$.

The support graph of inequality (15) for the case in which $t - i + 1 \geq i$ is depicted in Figure 1. First observe that at most one variable corresponding to a “left” (“right”) edge can take value 1 because the corresponding job cannot be assigned more than once. Now, if two variables corresponding to a left and a right edge take value 1, say $x_{is}$ and $y_{t-i+1,s'}$ (or $y_{js}$ and $x_{t-j+1,s'}$), then the time-slots before $\max\{s, s'\}$ are not sufficiently many to schedule all the jobs preceding $i \in J_1$ ($j \in J_2$) and $t - i + 1 \in J_2$ ($t - j + 1 \in J_1$).

**Theorem 3.4 Inequalities**

12
Figure 1: The support graph of inequalities (15) for the case in which \( t - i + 1 \geq i \).

\[
x_{it} + \sum_{s=i+1}^{t} x_{i+1,s} + y_{t-i,t} + \sum_{s=t+i+1}^{t} y_{t-i+1,s} \leq 1 \tag{17}
\]

for \( 1 \leq i \leq n_1 \) and \( t > i + 1 \), and

\[
y_{jt} + \sum_{s=j+1}^{i} y_{j+1,s} + x_{t-j,t} + \sum_{s=t-j+1}^{i} x_{t-j+1,s} \leq 1 \tag{18}
\]

for \( 1 \leq j \leq n_2 \) and \( t > j + 1 \), are valid and induce facets of \( P_{\text{PS}}^2 \).

Here again at most one \( x \)-variable (\( y \)-variable) occurring in (17) or (18) can take value 1. Furthermore, if one \( x \)-variable and one \( y \)-variable take value 1, then there are not enough time-slots to schedule the previous jobs. The support graph of inequality (17) for the case in which \( t - i + 1 > i + 1 \) is depicted in Figure 2.
Figure 2: The support graph of inequalities (17) for the case in which $t - i + 1 > i + 1$.

**Theorem 3.5 Inequalities**

$$\sum_{j=i}^{\min\{t,n_1\}} x_{jt} + \sum_{j=1}^{t-1} y_{jt} + \sum_{s=t-i+1}^{t} y_{t-i+1,s} \leq 1 \quad (19)$$

for $2 \leq i \leq n_1$ and $1 \leq t - i < n_2$, and

$$\sum_{i=j}^{\min\{t,n_2\}} y_{it} + \sum_{i=1}^{t-j} x_{it} + \sum_{s=t-j+1}^{t} x_{t-j+1,s} \leq 1 \quad (20)$$

for $2 \leq j \leq n_2$ and $1 \leq t - j < n_1$, are valid and induce facets of $P_{FS}^2$.

The interpretation of these inequalities is similar to that of the previous ones.
3.2 Odd hole inequalities

The next family of facet-inducing inequalities has the form

\[ x(S_1) + y(S_2) \leq 2 \]

and derives from odd holes of the conflict graph \( G \). Observe that in \( \text{PSP}_2 \) an incompatibility may arise not only between variables \( x_{is} \) and \( x_{jt} \), or \( y_{is} \) and \( y_{jt} \), but also between \( x_{is} \) and \( y_{jt} \), precisely when either trivially \( s = t \), or

\[ \max\{s, t\} < i + j - 1. \]

Consider then the 5-holes of \( G \) having the form

\[ H' = \{x_{it}, x_{i,t+1}, x_{i+1,t+2}, y_{t-i,t}, y_{t-i,t+2}\}, \]
\[ H'' = \{y_{js}, y_{j,s+1}, y_{j+1,s+2}, x_{s-j,s}, x_{s-j,s+2}\}, \]

for \( 1 \leq i < n_1 - 1, i < t < n - 1, 1 \leq j < n_2 - 1 \) and \( j < s < n - 1 \). Hence, inequalities

\[ x_{it} + x_{i,t+1} + x_{i+1,t+2} + y_{t-i,t} + y_{t-i,t+2} \leq 2 \]
\[ y_{js} + y_{j,s+1} + y_{j+1,s+2} + x_{s-j,s} + x_{s-j,s+2} \leq 2 \]

are valid for \( \text{PSP}_2 \). Let then

\[ f'(i, t) = x_{it} + x_{i,t+1} + x_{i+1,t+2} + y_{t-i,t} + y_{t-i,t+2}, \]
\[ f''(j, s) = y_{js} + y_{j,s+1} + y_{j+1,s+2} + x_{s-j,s} + x_{s-j,s+2}, \]

for all \( 1 \leq i < n_1 - 1, i < t < n - 1, 1 \leq j < n_2 - 1 \) and \( j < s < n - 1 \). We obtain constraints (21) and (22) by lifting the 5-hole inequalities.

**Theorem 3.6 Inequalities**

\[ f'(i, t) + \sum_{s=t+1}^{t} x_{i+1,s} + \sum_{j=t+3}^{t} x_{j,t+2} + \sum_{j=t-i+1}^{t-i+2} \sum_{s=j}^{\min\{t-i+1,n_1\}} y_{js} + \sum_{j=t-n_1+2}^{\min\{t-i+1,n_2\}} y_{j,t+2} \leq 2 \]

(21)
for $1 \leq i < n_1 - 1$ and $i < t < n - 1$, and

$$f''(j, s) + \sum_{h=j+1}^{s} y_{j+h} + \sum_{h=j+3}^{\min\{s+2,n_2\}} y_{h,s+2} + \sum_{i=s-j+1}^{\min\{s-j+1,n_1\}} \sum_{h=i}^{s+1} x_{ih} + \sum_{h=s-n_2+2}^{\min\{s-j-1,n_1\}} x_{h,s+2} \leq 2$$ (22)

for $1 \leq j < n_2 - 1$ and $j < s < n_1 - 1$, are valid and induce facets of $P_{PS}^2$.

Proof. See [1].

Finding the general form of all facet-defining inequalities corresponding to maximal cliques of the conflict graph $G$ and those obtained by lifting odd-holes inequalities is an open problem. Observe that no subgraph of $G$ induced by vertices corresponding to $x$ (to $y$) variables only, contains an odd hole.

### 3.3 Other inequalities

The following facet-inducing inequalities do not correspond to subgraphs of the conflict graph, and have the form

$$-x_{it} + x(S_1) + y(S_2) \leq 1 \quad \text{or} \quad -y_{jt} + x(S_1) + y(S_2) \leq 1$$

for suitable values of $t \in T$ and any $1 \leq i \leq n_1$, $1 \leq j \leq n_2$.

**Theorem 3.7 Inequalities**

$$-x_{it} + \sum_{h=t-i}^{t-1} y_{t-i+h} + y_{t-i,s} + x_{s-t+i,s} + \sum_{h=s-t+i+1}^{s} x_{s-t+i+1,h} \leq 1$$ (23)

for $1 \leq i < n_1$, $t > i + 1$ and $s > t$, and

$$-y_{jt} + \sum_{h=t-j}^{t-1} x_{t-j,h} + x_{t-j,s} + y_{s-t+j,s} + \sum_{h=s-t+j+1}^{s} y_{s-t+j+1,h} \leq 1$$ (24)

for $1 \leq j < n_2$, $t > j + 1$ and $s > t$, are valid and induce facets of $P_{PS}^2$.

Again, no more than one $x$-variable ($y$-variable) with coefficient 1 can take value 1. Furthermore, if $x_{i'k} = 0$ ($y_{j's} = 0$) but there exists $t' < t$ such that $x_{i't'} = 1$ ($y_{j't'} = 1$), then the remaining time-slots are not sufficient to schedule one job of each chain having an assignment variable appearing in the constraints plus all the previous jobs. Finally, if $\sum_{s=1}^{t'} x_{is} = 0$ ($\sum_{s=1}^{t'} y_{js} = 0$), then the precedence
constraints imply that no job of $J_1$ (of $J_2$) with a variable occurring in the constraint can be scheduled. The support graph of inequality (23) is depicted in Figure 3.

Proof. See [1].

![Figure 3: The support graph of inequalities (23).](image)

3.4 A corollary: PSP$_1$

Theorems 2.2 and 3.1 (inequality (12)) clearly provide facets of $P_{PS}^1$.

Observation 3.1 For $m = 1$, inequalities (3) are redundant.

Proof. Writing (2) for PSP$_1$ and $j = 1$ and (5) we get

$$1 \geq \sum_{s=1}^{n} x_{1s} \geq \sum_{s=1}^{t} x_{1s}. $$
for any \( t \leq n \). Let \( r = \min\{t, n_1\} \). Then by (5) and (7),

\[
1 \geq \sum_{s=1}^{t} x_{1t} \geq x_{1t} + \sum_{s=2}^{t} x_{2s} \geq x_{1t} + x_{2t} + \sum_{s=3}^{t} x_{3s} \geq \cdots \geq \sum_{j=1}^{r} x_{jt} + \sum_{s=r}^{t-1} x_{n_1s} \geq \sum_{j=1}^{r} x_{jt}
\]

where \( \sum_{s=r}^{t-1} x_{n_1s} = 0 \) for \( r = t \).

We can prove that

**Theorem 3.8** Let \( Q \) be the polytope defined by (12) and by (9), (10) written for \( m = 1 \). Then \( Q = P_{PS}^1 \).

**Proof.** Clearly \( Q \supseteq P_{PS}^1 \). Let \( x^* \) be a fractional point of \( Q \). We show that \( x^* \) is not an extreme point of \( Q \). For all \( 1 \leq j \leq n_1 \), let

\[
t(j) = \min\{t : x^*_{jt} > 0\}.
\]

From precedence constraints (12), it follows that \( i < j \) implies \( t(i) < t(j) \).

Let \( y \) be defined as

\[
y_{jt(j)} = \left[x^*_{jt(j)}\right], \quad y_{jt} = 0 \quad \text{for all } t \neq t(j)
\]

for \( 1 \leq j \leq n_1 \) and \( t \geq j \). It is immediately seen that \( y \in Q \).

Define then \( z \) as

\[
z_{jt} = \frac{x^*_{jt(j)}}{1 - \lambda} - \frac{\lambda}{1 - \lambda}, \quad z_{jt} = \frac{x^*_{jt}}{1 - \lambda} \quad \text{for all } t \neq t(j),
\]

for \( 1 \leq j \leq n_1 \), \( t \geq j \) and for \( 0 < \lambda < 1 \), \( \lambda = \min\{x^*_{jt} : x^*_{jt} > 0\} \). Let us show that also \( z \in Q \).

\( i \) Trivially, \( z \) satisfies non-negativity constraints (10).

\( ii \) \( z \) satisfies (9): since \( x^* \in Q \), one has

\[
\sum_{s=1}^{n} z_{1s} = z_{1t(1)} + \sum_{s \neq t(1)} z_{1s} = \frac{x^*_{1t(1)}}{1 - \lambda} - \frac{\lambda}{1 - \lambda} + \sum_{s \neq t(1)} x^*_{1s} = \frac{\sum_{s=1}^{n} x^*_{1s} - \lambda}{1 - \lambda} \leq 1.
\]

\( iii \) \( z \) satisfies (12). Indeed,

\[
\sum_{s=j-1}^{t-1} z_{j-1,s} = \begin{cases} 
\sum_{s=j-1}^{t-1} \frac{x^*_{j-1,s} - \lambda}{1 - \lambda} & \text{if } t(j - 1) \leq t - 1 \\
\sum_{s=j-1}^{t-1} \frac{x^*_{j-1,s}}{1 - \lambda} & \text{if } t(j - 1) > t - 1
\end{cases}
\]

where
and, similarly,

\[
\sum_{s=j}^t z_{is} = \begin{cases} 
\frac{\sum_{s=j}^t x_{is} - \lambda}{1 - \lambda} & \text{if } t(j) \leq t \\
\sum_{s=j}^t x_{is} & \text{if } t(j) > t.
\end{cases}
\]

Using equalities (a)-(d) and the fact that if \( t(j) \leq t \) \( \sum_{s=t-1}^{t-1} x_{i-1,t} \geq \sum_{s=1}^{t} x_{i,s} \), one concludes that \( z \) satisfies (12).

The proof is completed by observing that \( \mathbf{x}^* = \lambda \mathbf{y} + (1 - \lambda) \mathbf{z} \).

4 CSP$_2$ and ShP$_2$

Unlike Section 3, the study of the polytope is here based on a dynamic programming algorithm that finds an optimal solution; see Wolsey [22] and Martin et al. [10]. An algorithm of this type, available for PSP$_m$, CSP$_m$, ShP$_m$ for any given \( m \), allows us to transform each of these problems to a special shortest path problem. Time-indexed variables correspond then to (groups of) variables of the shortest path problem, and facet-inducing inequalities can be obtained by suitable projections eliminating the unnecessary variables.

In principle, the method exposed can be applied to any of the three problems. However, the shortest path reformulation of PSP$_2$ includes a larger number of variables than those of CSP$_2$ and ShP$_2$, and the projection turns out to be much more complicated. On the other hand, the method is particularly suitable for CSP$_2$ and ShP$_2$, because the relevant polyhedra are not full-dimensional.

4.1 CSP$_2$

Let us first consider the case \( n > n_1 + n_2 \) (CSP$_2$). Observe that, in this case variable \( x_{it} \) (\( y_{js} \)) is defined if and only if \( i \leq t \leq i + n - n_1 \) \( j \leq s \leq j + n - n_2 \).

Consider an optimal assignment of exactly \( i \) jobs in \( J_1 \) and \( j \) jobs in \( J_2 \) to \( t \geq i + j \) slots. The corresponding utility \( f(i, j, t) \) can be computed in \( \mathcal{O}(n_1 n_2 n) \) time using the following recursion:

\[
f(i, j, t) = \max \{ f(i-1, j, t-1) + f_1^i(t), \quad f(i, j-1, t-1) + f_2^j(t), \quad f(i, j, t-1) \}
\]

(25)

under the initial condition \( f(0, 0, t) = 0 \) for all \( t \in T \).

Associate a variable \( z_{ijt} \) with each \( f(i, j, t) \) in equation (25). An optimal solution \( z^* \) to the following linear program
\[
\begin{align*}
\min z_{n_1n_2n} \\
z_{ijs} - z_{i-1,j,s-1} & \geq a_{is}, \quad i \geq 1, j \geq 0, i + j \leq s \leq \sigma(i,j) \\
z_{ijs} - z_{i,j-1,s-1} & \geq b_{js}, \quad i \geq 0, j \geq 1, i + j \leq s \leq \sigma(i,j) \\
z_{ijs} - z_{i,j,s-1} & \geq 0, \quad i \geq 0, j \geq 0, i + j < s \leq \sigma(i,j)
\end{align*}
\]

\[\text{gives } z_{ijs}^* = f(i,j,t) \text{ for any } i,j,t, \text{ where } \sigma(i,j) = n - (n_1 - i) - (n_2 - j). \text{ Note that the constraint matrix of (26) is totally unimodular. Let } w_{ijs}, v_{ijs} \text{ and } u_{ijs} \text{ be the dual variables associated with the constraints of (26). For } 1 \leq i \leq n_1, 1 \leq j \leq n_2 \text{ and } s \geq i + j, \text{ the dual is}
\]

\[
\begin{align*}
\max \left( & \sum_{i=1}^{n_1} \sum_{s=i}^{i+n-n_1} a_{is} \sum_{j=\max\{0,s-\sigma(i,0)\}}^{\min\{n_2,s-i\}} w_{ijs} + \\
& + \sum_{j=1}^{n_2} \sum_{s=j}^{j+n-n_2} b_{js} \sum_{i=\max\{0,s-\sigma(0,j)\}}^{\min\{n_1,s-j\}} v_{ijs} \right)
\end{align*}
\]
Comparing (27) to the objective function of the original problem, one obtains

\[
\begin{align*}
    & w_{0s} + u_{0s} - w_{i+1,0,s+1} - v_{i,s+1} - u_{i0,s+1} = 0 & (28) \\
    & v_{0js} + u_{0js} - w_{1,j+1,s+1} - v_{0,j+1,s+1} - u_{0j,s+1} = 0 & (29) \\
    & w_{ijs} + v_{ijs} + u_{ijs} - w_{i+1,j,s+1} - v_{i,j+1,s+1} - u_{i,j,s+1} = 0 & (30) \\
    & w_{n1n2s} + v_{n1n2s} + u_{n1n2,s} - u_{n2s} = 0 & (31) \\
    & w_{n1n2n} + v_{n1n2n} + u_{n1n2n} = 1 & (32) \\
    & w_{ijs}, v_{ijs} \geq 0 & (33) \\
    & u_{i+s}^j \geq 0 & (34)
\end{align*}
\]

for any \(1 \leq i \leq n_1\) and \(1 \leq j \leq n_2\), and since the polyhedron (28)-(34) is integrable, \(P_{CS}^2 = \text{proj}_{x,y}((x,y,u,v,w) : (28)-(36))\), where \(\text{proj}_{x,y}(S)\) represents the projection of \(S\) on the \((x,y)\)-space.

**Projection 1**

1. Rewrite dual constraints (28) for \(i = n_1\) and \(n_1 \leq s \leq \sigma(n_1,0)\), and (29) for \(i = n_1, j \geq 1\) and \(n_1 + j \leq s \leq \sigma(n_1,j)\):

\[
    w_{n10s} - v_{n11,s+1} - u_{n20,n1+1} = 0
\]
\[ w_{n_1, 0, n_1+1} + u_{n_1, 0, n_1+1} - v_{n_1, 1, n_1+2} - u_{n_1, 0, n_1+2} = 0 \]
\[ w_{n_1, 1, n_1+1} + v_{n_1, 1, n_1+1} - v_{n_1, 2, n_1+2} - u_{n_1, 1, n_1+2} = 0 \]
\[ \ldots \]
\[ \ldots \]
\[ w_{n_1, 0, n-n_2} + u_{n_1, 0, n-n_2} = 0 \]
\[ w_{n_1, 1, n-(n_2-1)} + v_{n_1, 1, n-(n_2-1)} + u_{n_1, 1, n-(n_2-1)} - v_{n_1, 2, n-n_2+2} = 0 \]
\[ \ldots \]
\[ w_{n_1, n_2-1, n-1} + v_{n_1, n_2-1, n-1} + u_{n_1, n_2-1, n-1} - v_{n_1, n_2} = 0 \]
\[ w_{n_1, n_2} + v_{n_1, n_2} + u_{n_1, n_2} = 1 \]

2. Summing up the above equations, we get
\[
\sum_{s=n_1}^{n} \min\{n_2, s-n_1\} \sum_{j=\max\{0, s-\sigma(n_1, 0)\}}^{\min\{n_2, s-n_1\}} w_{n_1js} = 1,
\]
from which
\[
\sum_{s=n_1}^{n} x_{n_1s} = 1
\]
can be obtained exploiting (35).

3. Repeating the same procedure for \(1 \leq i \leq n_1 - 1, j \geq 1\) and \(i + j \leq s \leq \sigma(i, j)\) one obtains
\[
\sum_{s=i}^{i+n_1-1} \min\{n_2, s-i\} \sum_{j=\max\{0, s-\sigma(i, 0)\}}^{\min\{n_2, s-i\}} w_{ijs} = \sum_{s=i+1}^{i+1+n_1-1} \min\{n_2, s-i-1\} \sum_{j=\max\{0, s-\sigma(0, j)\}}^{\min\{n_2, s-i-1\}} w_{i+1js}
\]
from which equations (8) for \(k = 1\) can again be obtained exploiting (35). Equations (8) for \(k = 2\) are derived in a similar way.

**Projection 2**

1. For \(1 \leq i \leq n_1 - 1\) and \(i \leq t \leq \sigma(i, 0)\), rewrite dual constraints (28) for \(i \leq s \leq t\), and (29) for \(i + j \leq s \leq t\):
\[
w_{i0i} - w_{i+1, 0, i+1} - v_{i+1, i+1} - u_{i0, i+1} = 0 \quad (37)
\]
2. From equation (37), one gets

\[ w_{i0} = w_{i+1,0} + v_{i+1,1} + u_{i+1} \]

from which

\[ x_{ii} \geq x_{i+1,1+1} \]

can be obtained exploiting (33), (34) and (35).

3. Summing up equations (37) and (38), one gets

\[
\sum_{s=i}^{i+1} \sum_{j=0}^{s-i} w_{ij} = \sum_{s=i+1}^{i+2} \sum_{j=0}^{s-i-1} w_{i+1,js} + \sum_{j=0}^{1} (v_{ij+1,i+2} + u_{ij,i+2})
\]

from which

\[ x_{ii} + x_{i,1+1} \geq x_{i+1,i+1} + x_{i+1,i+2} \]

can be derived.

In general, iterating the above procedure one obtains equation

\[
\sum_{s=1}^{t} \sum_{j=\max(0,s-\sigma(i,0))}^{\min(n_2,s-1)} w_{ijs} = \sum_{s=i+1}^{t+1} \sum_{j=\max(0,s-\sigma(i+1,0))}^{\min(n_2,s-i-1)} w_{i+1,js} + \sum_{j=\max(0,s-\sigma(i+1,0))}^{\min(t-i,n_2)} (v_{ij+1,i+1} + u_{ij,i+1})
\]

from which inequalities (12) can as usual be obtained exploiting (33), (34) and (35).

Precedence constraints (13) can be derived in a similar way.

**Projection 3**
1. From dual constraint (32), one gets
\[ w_{n_1 n_2 n} + v_{n_1 n_2 n} \leq 1. \]

Exploiting (34), (35) and (36), immediately one has
\[ x_{n_1 n} + y_{n_2 n} \leq 1. \]

2. Rewriting dual constraints (30) for \( s = n - 1, i = n_1 - 1 \) and \( n_1 \) and max\{0, s - \( \sigma(i, 0) \)\} \( \leq j \leq \min\{n_2, s - i\} \) and summing up them, one gets
\[
\sum_{i=n_1-1}^{n_1-1} \sum_{j=n_2+n_1-i-1}^{n_2} w_{ij, n-1} + \sum_{j=n_2-1}^{n_2} \sum_{i=n_1+n_2-j-1}^{n_1} v_{ij, n-1} + \\
\sum_{i=n_1-1}^{n_1} \sum_{j=n_2+n_1-i-1}^{n_2} u_{ij, n-1} = w_{n_1 n_2 n} + v_{n_1 n_2 n} + u_{n_1 n_2 n} = 1.
\]

Exploiting again (34), (35) and (36), one obtains
\[
\sum_{i=n_1-1}^{n_1-1} x_{i, n-1} + \sum_{j=n_2-1}^{n_2} y_{j, n-1} \leq 1.
\]

In general, iterating the above procedure for \( s < n - 1, s + n_1 - n \leq i \leq n_1 \) and max\{0, s - \( \sigma(i, 0) \)\} \( \leq j \leq \min\{n_2, s - i\} \) one obtains the inequalities
\[
\sum_{i=\max\{1, i+n_1-n\}}^{\min\{n_1, t\}} x_{it} + \sum_{j=\max\{1, t+n_2-n\}}^{\min\{n_2, t\}} y_{jt} \leq 1
\]
i.e., the slot assignment constraints for problem CSP\(_2\).

**Theorem 4.1** Job assignment constraints (8), precedence constraints (7) and slot assignment constraints (3) induce facets of \( P_{CS}^2 \).

### 4.2 ShP\(_2\)

Let us now consider the case \( n = n_1 + n_2 \) (ShP\(_2\)). In this case \( x_{it} (y_{jt}) \) is defined only for \( i \leq t \leq i + n_2 \) \( (j \leq t \leq j + n_1) \). Suppose that in an optimal assignment of \( t \leq n \) slots, \( j \) slots are assigned to (the first \( j \) jobs of) \( J_2 \) and the remaining ones to \( J_1 \). The corresponding utility \( f(j, t) \) is then given by the recursion:
\[
f(j, t) = \max\{f(j, t - 1) + f^1_{t-j}(t); f(j - 1, t - 1) + f^2_j(t)\}, \quad (40)
\]
under the initial condition $f(0, 0) = 0$. Observe that $f(j, t)$ is defined for $j \leq t \leq j + n_1$ and the computation requires $O(n^2)$ time. Formula (40) can be extended to PSP$_2$ and CSP$_2$ if $f^k_j(t)$ is non-negative regular (i.e., non-increasing with respect to $t$).

The solutions to ShP$_2$ are therefore in a one-to-one correspondence with $(s, t)$-paths of a directed acyclic grid $G$ constructed as follows (see Figure 4):

- there exists a node $(j, t)$ for any $0 \leq j \leq n_2$ and $j \leq t \leq j + n_1$;
- $s = (0, 0), t = (n_2, n)$;
- the “horizontal” edges have the form $(j - 1, t - 1) \rightarrow (j, t)$ and are weighted by $b_{jt}$;
- the “vertical” edges have the form $(j, t - 1) \rightarrow (j, t)$ and are weighted by $a_{t-j,t}$.

Since $G$ is acyclic, the convex hull of its $(s, t)$-paths and thus also the complete description of $F_{ShP}^2$ is $\{z \in \mathbb{R}^d | Ez = e_t, z \geq 0\}$, where $E$ is the node-arc incidence matrix of $G$ from which the row corresponding to $s = (0, 0)$ has been removed, and $e_t$ is the unit vector having value 1 in position $t$. 

![Figure 4: Directed acyclic graph (grid).](image-url)
5 Conclusions

This paper addressed a one-machine scheduling problem in which \( m \) disjoint sets of jobs can be assigned to \( n \) time-slots under chain-like precedence constraints. Two variants of the problem have been considered, depending on whether a feasible schedule must be complete, i.e., each job must be assigned to one time-slot (Complete Scheduling Problem, CSP\(_m\)), or not (Partial Scheduling Problem, PSP\(_m\)). A variant of CSP\(_m\) (Shuffling Problem, ShP\(_m\)), when \( m > 1 \) and \( n \) equals the total number of jobs in the system, has also been considered. We defined the polyhedra \( P_{PS}^m, P_{CS}^m \) and \( P_{Sh}^m \), respectively, as the convex hulls of feasible solutions to PSP\(_m\), CSP\(_m\) and ShP\(_m\), and provided a partial description of \( P_{PS}^2 \) and \( P_{CS}^2 \). Moreover, \( P_{Sh}^m \) turns out to be the convex hull of \((s,t)\)-paths on a special directed acyclic grid.

Further research effort should be directed towards

- a complete description of \( P_{PS}^2 \) and \( C_{PS}^2 \), perhaps along the lines suggested by experiments with PORTA [15];
- an extension to \( m > 2 \) of the results presented here.

A relation has been highlighted between the polytopes studied and the stable set polytope of the conflict graph \( G \) associated with the problem. It turns out that \( P_{PS}^m \subseteq \text{Stab}(G) \), and some facet-inducing inequalities of the latter are also facets of the former. Under what general conditions a facet of \( \text{Stab}(G) \) can be exported to a facet of \( P_{PS}^m \) is then an interesting open question.

Acknowledgement

The authors gratefully acknowledge the Editor and two anonymous Referees for useful comments and indications.

References


http://www2.iwr.uni-heidelberg.de/groups/comopt/software/PORTA/


