The Flowshop Scheduling Polyhedron with Setup Times[1]

Roger Z. Ríos-Mercado[2]
Graduate Program in Operations Research
University of Texas at Austin
Austin, TX 78712–1063
roger@bajor.me.utexas.edu

Jonathan F. Bard[3]
Graduate Program in Operations Research
University of Texas at Austin
Austin, TX 78712–1063
jbard@mail.utexas.edu

July 1996

[1]Technical Report ORP96–07, Graduate Program in Operations Research, University of Texas at Austin, Austin, TX 78712–1063

[2]Research partially supported by the Mexican National Council of Science and Technology (CONACYT) and by an E. D. Farmer fellowship from The University of Texas at Austin

[3]Research partially supported by a grant from the Texas Higher Education Coordinating Board under the Advanced Research Program, ARP 003658–003

Abstract

This paper presents the polyhedral structure of two different mixed-integer programming representations of the flowshop scheduling problem with sequence-dependent setup times, denoted by SDST flowshop. The first is related to the asymmetric traveling salesman problem polytope. The second is less common and is derived from a model proposed by Srikar and Ghosh, giving what we call the S-G polytope. It is shown that any facet-defining inequality (facet) of either of these polytopes induces a facet for the SDST flowshop polyhedron. Facets for the S-G polytope and valid mixed-integer inequalities based on variable upper-bound flow inequalities for either formulation are developed as well.

Keywords: flowshop scheduling, setup times, polyhedral combinatorics, facet-defining inequalities, asymmetric traveling salesman problem

1 Introduction

In this paper, we address the problem of finding a permutation schedule of n jobs in an m-machine flowshop environment that minimizes the maximum completion time $C_{\max}$ of all jobs, also known as the makespan. The jobs are available at time zero and have sequence-dependent setup times on each machine. All parameters, such as processing and setup times, are assumed to be known with certainty. This problem is regarded in the scheduling literature as the sequence-dependent setup time flowshop (SDST flowshop) and is evidently $\mathcal{NP}$-hard since the case where $m = 1$ is simply a traveling salesman problem (TSP).

Applications of sequence-dependent scheduling are commonly found in most manufacturing environments. In the printing industry, for example, presses must be cleaned and settings changed when ink color, paper size or receiving medium differ from one job to the next. Setup times are strongly dependent on the job order. In the container manufacturing industry machines must be adjusted whenever the dimensions of the containers are changed, while in printed circuit board assembly, rearranging and restocking component inventories on the magazine rack is required between batches. In each of these situations, sequence-dependent setup times play a major role and must be considered explicitly when modeling the problem.

The objective of this paper is to study the SDST flowshop polyhedron; i.e., the convex hull of incidence vectors of all feasible solutions. In so doing, we consider two different models or formulations. Model A is based on the asymmetric traveling salesman problem (ATSP) and model B is based on a formulation due to Srikar and Ghosh [14]. In each case, two sets of variables are identified: a set of binary decision variables which determines the sequence or ordering of the jobs, and a set of nonnegative real variables which determines the times processing begins for each job. When the time variables are ignored the binary variables give rise to a subspace of the SDST flowshop consisting of the convex hull of incidence vectors of feasible sequences. For model A, this subspace is the well known ATSP polytope; for model B, the corresponding subspace (here, called the S-G polytope) has not been previously studied. In our work, we show how any facet-defining inequality (or facet) for either of these polytopes induces a facet for the SDST flowshop polyhedron. We also investigate the facial structure of the S-G polytope and develop several valid inequalities for the SDST flowshop polyhedron.

The rest of the paper is organized as follows. In Section 2 we introduce the mathematical models A and B, and discuss their basic differences. A brief literature review is presented in Section 3. This is followed in Section 4 with some background material on polyhedral theory. Major results relating to the polyhedral structure of models A and B are given in Sections 5 and 6, respectively.

2 Mathematical Formulation

In the flowshop environment, a set of n jobs must be scheduled on a set of m machines, where each job has the same routing. Therefore, without loss of generality, we assume that the machines are ordered according to how they are visited by each job. Although for a general flowshop the job sequence may not be the same for every machine, here we assume a *permutation schedule*; i.e., a subset of the feasible schedules that requires the same job sequence on every machine. We suppose that each job is available at time zero and has no due date. We also assume that there is a setup time which is sequence dependent so that for every machine i there is a setup time that must precede the start of a given task that depends on both the job to be processed (k) and the job that immediately precedes it (j). The setup time on machine i is denoted by s_{ijk} and is assumed to be *asymmetric*; i.e., $s_{ijk} \neq s_{ikj}$. After the last job has been processed on a given machine, the machine is brought back to an acceptable "ending" state. We assume that this last operation takes zero time because we are interested in job completion time rather than machine completion time. Our objective is to minimize the time at which the last job in the sequence finishes processing on the last machine, also known as *makespan*. This problem is denoted by $Fm|s_{ijk},\ prmu|C_{\max}$ or SDST flowshop.

In modeling this problem as a mixed integer program (MIP), we consider two different formulations. In the first case, a set of the binary variables is used to define whether or not one job is an immediate predecessor of another; in the second case, the binary variables simply determine whether or not one job precedes another. A set of nonnegative real variables is also included in the formulations. In either case they have the same definition and are used to determine the starting time of each job on each machine.

Triangle inequality: The triangle inequality for the setup times is stated as follows:

$$s_{ijk} + s_{ikl} \geq s_{ijl} \quad \text{for } i \in I,\ j,k,l \in J. \tag{1}$$

Throughout the sequel, we will assume that the triangle inequality holds unless otherwise stated. In most operations (e.g., see [14, 15]), the time it takes to set up a machine from job j to job l is less than the time it takes to set up a machine from j to another job k, and then set up the machine from k to l. Nevertheless, if there really exists a machine i and jobs j, k, l such that $s_{ijk} + s_{ikl} < s_{ijl}$, we can always replace s_{ijl} with $s'_{ijl} = s_{ijk} + s_{ikl}$ and force (1) to hold as an equality.

2.1 Notation

In the development of the mathematical model, we make use of the following notation.

Indices and sets

m number of machines

n number of jobs

i machine index; $i \in I = \{1, 2, \ldots, m\}$

j, k, l job indices; $j, k, l \in J = \{1, 2, \ldots, n\}$

J_0 $= J \cup \{0\}$ extended set of jobs, including a dummy job denoted by 0

Input data

p_{ij} processing time of job j on machine i; $i \in I$, $j \in J$

s_{ijk} setup time on machine i when job j is scheduled right before job k; $i \in I$, $j \in J_0$, $k \in J$

Computed parameters

A_i upper bound on the time at which machine i finishes processing its last job; $i \in I$,

$$A_i = A_{i-1} + \sum_{j \in J} p_{ij} + \min \left\{ \sum_{j \in J_0} \max_{k \in J} \{s_{ijk}\}, \sum_{k \in J} \max_{j \in J_0} \{s_{ijk}\} \right\}$$

where $A_0 = 0$

B_{ij} lower bound on the starting time of job j on machine i; $i \in I$, $j \in J$

$$B_{ij} = \max \{s_{i0j}, B_{i-1,j} + p_{i-1,j}\} \qquad i \in I, j \in J$$

where $B_{0j} = 0$ for all $j \in J$

Common variables

y_{ij} nonnegative real variable equal to the starting time of job j on machine i; $i \in I$, $j \in J$

$C_{\max}$ nonnegative real variable equal to the makespan;

$$C_{\max} = \max_{j \in J} \{y_{mj} + p_{mj}\}$$

2.2 Formulation A

Let $A = \{(j, k) : j, k \in J_0, j \neq k\}$ the set of arcs in a complete directed graph induced by the node set J_0. We define the decision variables as follows:

$$x_{jk} = \begin{cases} 1 & \text{if job } j \text{ is the immediate predecessor of job } k;\ (j,k) \in A \\ 0 & \text{otherwise} \end{cases}$$

In the definition of x_{jk}, notice that $x_{0j} = 1$ ($x_{j0} = 1$) implies that job j is the first (last) job in the sequence for $j \in J$. Also notice that s_{i0k} denotes the initial setup time on machine i when job k has no predecessor; that is, when job k is scheduled first, for $k \in J$. This variable definition yields what we call a TSP-based formulation.

$$\text{Minimize} \quad C_{\max} \tag{2.1}$$

subject to

$$\sum_{j \in J_0} x_{jk} = 1 \qquad k \in J_0 \tag{2.2}$$

$$\sum_{k \in J_0} x_{jk} = 1 \qquad j \in J_0 \tag{2.3}$$

$$y_{ij} + p_{ij} + s_{ijk} \leq y_{ik} + A_i(1 \Leftrightarrow x_{jk}) \qquad i \in I,\ j,k \in J \tag{2.4}$$

$$y_{mj} + p_{mj} \leq C_{\max} \qquad j \in J \tag{2.5}$$

$$y_{ij} + p_{ij} \leq y_{i+1,j} \qquad i \in I \setminus \{m\},\ j \in J \tag{2.6}$$

$$x_{jk} \in \{0,1\} \qquad j,k \in J_0,\ j \neq k \tag{2.7}$$

$$y_{ij} \geq B_{ij} \qquad i \in I,\ j \in J \tag{2.8}$$

Equations (2.2) and (2.3) state that every job must have a predecessor and successor, respectively. Note that one of these $2n+2$ assignment constraints is redundant in the description of the feasible set. Time-based subtour elimination constraints are given by (2.4). This establishes that if job j precedes job k, then the starting time of job k on machine i must not exceed the completion time of job j on machine i ($y_{ij} + p_{ij}$) plus the corresponding setup time. Here, A_i is a large enough number (an upper bound on the completion time on machine i). Constraint (2.5) assures that the makespan is greater than or equal to the completion time of all jobs on the last machine, while (2.6) states that a job cannot start processing on one machine if it has not finished processing on the previous one. A lower bound on the starting time for each job on each machine is set in (2.8).

In formulation (2.1)-(2.8), we assume that s_{ij0}, the time required to bring machine i to an acceptable end state when job j is processed last, is zero for all $i \in I$. Thus the makespan is governed by the completion times of the jobs only. We are also assuming that all jobs need processing on all machines. If this last condition were not true, then eq. (2.5) could be replaced by

$$y_{ij} + p_{ij} \leq C_{\max} \qquad i \in I,\ j \in J$$

at the expense of increasing the number of makespan constraints from n to mn. Note that it is possible to combine $p_{ij} + s_{ijk}$ in (2.4) into a single term $t_{ijk} = p_{ij} + s_{ijk}$, but that we still need to handle the processing times p_{ij} separately in constraints (2.5) and (2.6).

If the triangle inequality does not hold, the lower bound constraint (2.8) must be replaced by

$$B_{ij} \leq y_{ij} + C_i(1 \Leftrightarrow x_{0j}) \qquad i \in I,\ j \in J,$$

where C_i is a large enough number (an upper bound on the initial setup time for machine i).

2.3 Formulation B

Srikar and Ghosh [14] proposed a second MIP formulation for $F|s_{ijk}, prmu|C_{\max}$. Their formulation contained a slight error that was later corrected by Stafford and Tseng [15]. The Srikar-Ghosh model does not consider the initial setup time s_{i0k} for the first job in the sequence, that is, it is assumed to be zero. Our formulation includes this parameter.

Let $\hat{A} = \{(j,k) : j,k \in J,\ j < k\}$. The decision variables are defined as follows:

$$x_{jk} = \begin{cases} 1 & \text{if job } j \text{ is scheduled any time before job } k;\ (j,k) \in \hat{A} \\ 0 & \text{otherwise} \end{cases}$$

The MIP formulation is

$$\text{Minimize} \quad C_{\max} \tag{3.1}$$

subject to

$$\begin{aligned}
y_{ij} + p_{ij} + s_{ijk} &\leq y_{ik} + A_i(1 \Leftrightarrow x_{jk}) && i \in I, (j,k) \in \hat{A} && (3.2)\\
y_{ik} + p_{ik} + s_{ikj} &\leq y_{ij} + A_i(x_{jk}) && i \in I, (j,k) \in \hat{A} && (3.3)\\
y_{mj} + p_{mj} &\leq C_{\max} && j \in J && (3.4)\\
y_{ij} + p_{ij} &\leq y_{i+1,j} && i \in I \setminus \{m\},\ j \in J && (3.5)\\
x_{jk} &\in \{0,1\} && (j,k) \in \hat{A} && (3.6)\\
y_{ij} &\geq B_{ij} && i \in I,\ j \in J && (3.7)
\end{aligned}$$

Constraints (3.2) and (3.3) ensure that time precedence is not violated. They also eliminate cycles. Equation (3.4) establishes the makespan criterion. Equation (3.5) states that a job cannot start processing on one machine if it has not finished processing on the previous machine. A lower bound on the starting time of each job on each machine is set in (3.7).

Srikar and Ghosh point out that the triangle inequality must hold in order for constraints (3.2)-(3.3) to hold. However, Stafford and Tseng provide a stronger condition for constraints (3.2)-(3.3) to be valid; i.e.,

$$s_{ijk} + s_{ikl} + p_{ik} \geq s_{ijl} \quad \text{for all } i \in I,\ j,k,l \in J. \tag{4}$$

Note that (4) is stronger than the triangle inequality (1), and implies that constraints (3.2)-(3.3) of the model hold, even if (1) does not hold for setup times. They illustrate this by means of an example.

If the triangle inequality does not hold, constraints (3.2), (3.3) and (3.7) are no longer valid. One possible replacement is

$$\begin{aligned}
y_{ij} + p_{ij} + s_{ijk} &\leq y_{ik} + (n+1)A_i(1 \Leftrightarrow x_{jk}) + A_i[P(k) \Leftrightarrow P(j) \Leftrightarrow 1] && i \in I,\ (j,k) \in \hat{A}\\
y_{ij} + p_{ij} + s_{ijk} &\leq y_{ik} + (n+1)A_i x_{jk} + A_i[P(j) \Leftrightarrow P(k) \Leftrightarrow 1] && i \in I,\ (j,k) \in \hat{A}\\
B_{ik} &\leq y_{ik} + C_i[P(k) \Leftrightarrow 1] && i \in I,\ k \in J,
\end{aligned}$$

respectively, where C_i is a large enough number (upper bound on the starting processing time of all jobs on machine i), and $P(j)$ represents the position in the schedule of job j, given by

$$P(j) \quad = \quad \sum_{p<j} x_{pj} + \sum_{q>j} (1 \Leftrightarrow x_{jq}) + 1 \quad j \in J. \tag{5}$$

In addition, the following constraints must be added to the formulation:

$$\begin{array}{lcll} P(j)+1 & \leq & P(k) + n(1 \Leftrightarrow x_{jk}) & (j,k) \in \hat{A} \\ P(k)+1 & \leq & P(j) + n x_{jk} & (j,k) \in \hat{A} \end{array}$$

Thus, when the triangle inequality does not hold, the problem size increases considerably.

2.4 Model Comparison

	Model A		Model B	
Variables	binary	$n(n+1)$	binary	$\frac{1}{2}n(n \Leftrightarrow 1)$
	real	$mn+1$	real	$mn+1$
	TOTAL	$n(n+1)+mn+1$	TOTAL	$\frac{1}{2}n(n \Leftrightarrow 1)+mn+1$
Constraints	(2.2)	$n+1$	(3.2)	$\frac{1}{2}mn(n \Leftrightarrow 1)$
	(2.3)	$n+1$	(3.3)	$\frac{1}{2}mn(n \Leftrightarrow 1)$
	(2.4)	$mn(n \Leftrightarrow 1)$		
	(2.5)	mn	(3.4)	mn
	(2.6)	$n(m \Leftrightarrow 1)$	(3.5)	$n(m \Leftrightarrow 1)$
	TOTAL	$mn^2+mn+n+2$	TOTAL	$mn^2+mn \Leftrightarrow n$
Nonzeros	(2.2)	$n(n+1)$	(3.2)	$\frac{3}{2}mn(n \Leftrightarrow 1)$
	(2.3)	$n(n+1)$	(3.3)	$\frac{3}{2}mn(n \Leftrightarrow 1)$
	(2.4)	$3mn(n \Leftrightarrow 1)$		
	(2.5)	$2mn$	(3.4)	$2mn$
	(2.6)	$2n(m \Leftrightarrow 1)$	(3.5)	$2n(m \Leftrightarrow 1)$
	TOTAL	$3mn^2+2n^2+mn$	TOTAL	$3mn^2+mn \Leftrightarrow 2n$

Table 1: *Problem size for models A and B*

Table 1 shows the problem size in terms of number of variables, constraints, and nonzeros for either model. As can be seen, model B is considerably smaller than model A in terms of both the number of constraints and the number of binary variables. This would appear to make it more attractive when considering exact enumeration methods such as branch-and-bound (B&B) and branch-and-cut (B&C). Nevertheless, the fact that much is known about the ATSP polytope gives added weight to model A. Table 2 displays the number of binary and real variables, number of constraints, number of nonzeros and density of the matrix of constraints for several values of m and n.

$m \times n$	model	binary	real	constraints	nonzeros	density
2×10	A	110	21	252	840	0.025
	B	45	21	230	620	0.041
2×20	A	420	41	902	3280	0.008
	B	190	41	860	2440	0.012
10×10	A	110	101	1212	3400	0.013
	B	45	101	1190	3180	0.018
10×20	A	420	201	4422	13200	0.005
	B	190	201	4380	12360	0.007

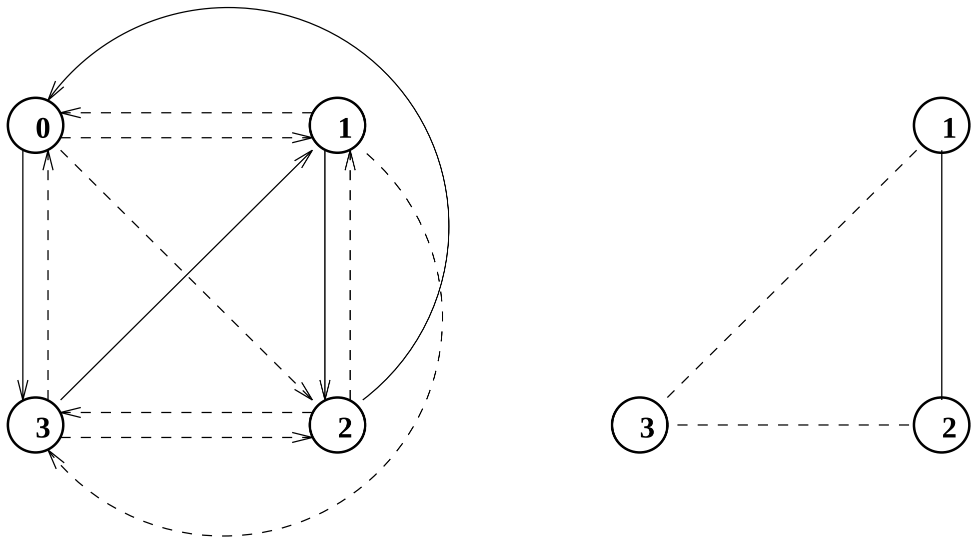
Table 2: *Problem size examples for models A and B*

To date, it has not been possible to tackle even moderate size instances of the SDST flowshop with either of these formulations due mainly to the weakness of their LP-relaxation lower bounds. LP-based enumeration procedures such as B&B and B&C require good LP-relaxation lower bounds. For example, Stafford and Tseng required about 6 hours of CPU time on a 80286-based PC to optimally solve a 5×7 instance using LINDO with formulation B. To improve the polyhedral representation of the relaxed feasible regions it is necessary to generate valid inequalities, the strongest being facets. One way to achieve this is by looking into the related subspaces: the ATSP polytope and the S-G polytope for models A and B, respectively. Many facets have been developed for the ATSP polytope over the last 20 years (e.g., see [1, 6, 2, 3, 11]). For model B, though, the S-G polytope remains unexplored. As we show presently, the facets of either of these polytopes can be extended to facets of the SDST flowshop polyhedron.

When comparing the ATSP polytope with the S-G polytope fundamental differences can be observed. In the former, we have a clear picture of what a feasible solution (also called a tour) looks like in a graph. This makes it easier to visualize, for instance, when certain constraints, such as the subtour eliminate constraints, may be violated. However, for model B, it is not a straightforward matter to identify in a graph a feasible solution from a given set of arcs. Figure 1 shows the graph for a 3-job problem and the solution for schedule $S = (3, 1, 2)$ for both models. For model B, an undirected graph can be used because x_{jk} is only defined for $j < k$. The dotted lines represent all feasible arcs (12 for model A and 3 for B); the solid lines identify the solution.

Figure 2 shows how a solution for model B can be built from a solution for model A. Note that each arc $\hat{e} \in \hat{A}$ (Step 2) is visited just once so the procedure is $O(|\hat{A}|) = O(n^2)$. In Step 1, a node (job) within brackets ($[j]$) denotes the job scheduled in the j-th position.

Likewise, a solution for model A can be easily constructed from a feasible solution for model B. Let $\hat{T}$ be an arc set representing a feasible schedule under model B. Let $\hat{x} \in B^{|\hat{A}|}$ be its corresponding characteristic vector; that is, $x_{jk} = 1$ if $(j, k) \in \hat{T}$, and $x_{jk} = 0$ otherwise. For each job j, its position



(a) Model A (b) Model B

Figure 1: *Graph representations for schedule (3,1,2)*

in the schedule $P(j)$ is determined by eq. (5) in $n(n-1)$ operations. The schedule S is found by sorting the jobs by increasing value of $P(j)$ and a feasible tour T is easily built from S in $O(n)$ time so that the complete conversion takes $O(n^2)$ time.

3 Related Research

We now highlight some previous work on the SDST flowshop and related problems.

Exact optimization: Formulation B was introduced by Srikar and Ghosh [14]. They used this model and the SCICONIC/VM mixed integer programming solver (based on branch-and-bound) to solve several randomly generated instances of the SDST flowshop. The largest solved was a 6-machine, 6-job problem, in about 22 minutes of CPU in a Prime 550 computer.

Later, Stafford and Tseng [15] corrected an error in the S-G formulation and using LINDO solved a 5×7 instance in about 6 hours of CPU on a PC. They also proposed three new MIP formulations of related flowshop problems based on the S-G model.

To the best of our knowledge, there have been no other attempts to solve the problem optimally using either formulation. However, Gupta [9] presents a branch-and-bound algorithm for the case where the objective is to minimize the total machine setup time. No computational results are reported. All other work has centered on variations of the 1- or 2-machine case, with the former being analogous to the TSP.

2-machine case: Work on $F2|s_{ijk}, prmu|C_{\max}$ includes Corwin and Esogbue [4], who consider a subclass of this problem that arises when one of the machines has no setup times. After estab-

Procedure `A-to-B()`

Input: An arc set T (or directed tour, $|T| = n+1$) corresponding to a feasible schedule for the SDST flowshop under model A.

Output: An arc set $\hat{T}$ corresponding to the equivalent schedule under model B.

Step 0. Initialize `mark(`$\hat{e}$`)=UNVISITED` for $\hat{e} \in \hat{A}$ and set $\hat{T} = \emptyset$
Step 1. Sort T as $T = \{(0,[1]),([1],[2]),\ldots,([n \Leftrightarrow 1],[n]),([n],0)\}$
Step 2. **for** $j = 1$ **to** n **do**
Step 2a. Choose the j-th arc in T
Step 2b. **for** each unvisited arc $\hat{e} \in \hat{A}$ incident to $[j]$ **do**
Step 2c. mark `mark(`$\hat{e}$`)=VISITED`
Step 2d. **for** each unvisited arc $\hat{e} \in \hat{A}$ incident from $[j]$ **do**
Step 2e. mark `mark(`$\hat{e}$`)=VISITED`
Step 2f. $\hat{T} \leftarrow \hat{T} + \hat{e}$
Step 3. Output $\hat{T}$

Figure 2: *Procedure to go from solution of A to solution of B*

lishing the optimality of permutation schedules, they develop an efficient dynamic programming formulation which they show is comparable, from a computational standpoint, to the corresponding formulation of the traveling salesman problem whose complexity is $O(n^2 2^n)$ [5]. No computations were performed.

Gupta and Darrow [8] establish the $\mathcal{NP}$-hardness of the problem and show that permutation schedules do not always minimize makespan. They derive sufficient conditions for a permutation schedule to be optimal, and propose and evaluate empirically four heuristics. They observe that the procedures perform quite well for problems where setup times are an order of magnitude smaller than processing times. However, when the magnitude of the setup times was in the same range as the processing times, the performance of the first two proposed algorithms decreased sharply.

Szwarc and Gupta [16] develop a polynomially bounded approximate method for the special case where the sequence-dependent setup times are additive. Their computational experiments on instances of up to 7 jobs show optimal results for the 2-machine case.

Heuristics: Most of the recent work for $F|s_{ijk},\ prmu|C_{\max}$ has focused on heuristics. Simons [13] describes four heuristics and compares them with three benchmarks that represent generally practiced approaches to scheduling in this environment. Experimental results for problems with up to

15 machines and 15 jobs are presented. His findings indicate that two of the proposed heuristics (`SETUP` and `TOTAL`) produce substantially better results than the other methods tested.

In [12], we developed a new greedy randomized adaptive search procedure (`GRASP`) and compared it to Simons' `SETUP` heuristic on a series of randomly generated problems of size up to 6×100. In the computations, `GRASP` outperformed `SETUP` on those instances where the setup times were an order of magnitude smaller than the processing times. When both parameters were identically distributed, `SETUP` was seen to be more effective.

4 Polyhedral Theory Background

The following definitions and well known theoretical results (e.g., see [10]) will be used in the developments.

A *polyhedron* $P \subseteq R^n$ is the set of points that satisfies a finite number of linear inequalities; i.e., $P = \{x \in R^n : Ax \le b\}$, where (A, b) is an $m \times (n+1)$ matrix. A polyhedron P is of *dimension* k, denoted $\dim(P) = k$, if the maximum number of affinely independent points in P is $k+1$. A polyhedron $P \subseteq R^n$ is *full-dimensional* if $\dim(P) = n$. Let $M = \{1, 2, \ldots, m\}, M^= = \{i \in M : a^i x = b_i \text{ for all } x \in P\}$ and let $M^\le = \{i \in M : a^i x < b_i \text{ for some } x \in P\} = M \setminus M^=$. Let $(A^=, b^=)$, $(A^\le, b^\le)$ be the corresponding rows of (A, b), referred as the *equality* and *inequality* sets of the representation (A, b) of P. A point $x \in P$ is called an *interior point* of P is $a^i x < b_i$ for all $i \in M$.

Lemma 1 *Let P be a polyhedron and let $(A^=, b^=)$ be its equality set. If $P \subseteq R^n$, then* $\dim(P) + \operatorname{rank}(A^=, b^=) = n$.

Corollary 1 *A polyhedron P is full-dimensional if and only if it has an interior point.*

The inequality $\pi x \le \pi_0$ [or (π, π_0)] is called a *valid inequality* for P if it is satisfied by all points in P. If (π, π_0) is a valid inequality for P and $F = \{x \in P : \pi x = \pi_0\}$, F is called a *face* of P, and we say that (π, π_0) *represents* F. A face F is said to be *proper* if $F \neq \emptyset$ and $F \neq P$. A face F of P is a *facet* of P if $\dim(F) = \dim(P) \Leftrightarrow 1$.

Theorem 1 *Let $(A^=, b^=)$ be the equality set of $P \subseteq R^n$ and let $F = \{x \in P : \pi x = \pi_0\}$ be a proper face of P, where $\pi \in R^n, \pi_0 \in R$. Then the following two statements are equivalent:*

(i) *F is a facet of P.*

(ii) *If $\lambda x = \lambda_0$ for all $x \in F$ then*

$$(\lambda, \lambda_0) = (\alpha\pi + uA^=, \alpha\pi_0 + ub^=)$$

for some $\alpha \in R$ and some $u \in R^{|M^=|}$.

Lemma 1 and Theorem 1 provide two different methods of characterizing facets of a polyhedron. We will also make use use of the following results on valid inequalities for variable upper-bound flow models to develop mixed-integer cuts.

Let

$$T = \{x \in B^n, z \in R_+^n \;:\; \sum_{j \in N^+} z_j \Leftrightarrow \sum_{j \in N^-} z_j \leq b,\; z_j \leq a_j x_j \quad \text{for } j \in N\} \tag{6}$$

where $N^+ \cup N^- = N$. Here $a_j \in R_+$ for $j \in N$ and $b \in R$. We say that $C \subseteq N^+$ is a *dependent set* if $\sum_{j \in C} a_j > b$.

Proposition 1 *If $C \subseteq N^+$ is a dependent set, $\lambda = \sum_{j \in C} a_j \Leftrightarrow b$, and $L \subseteq N^-$, then*

$$\sum_{j \in C} [z_j + (a_j \Leftrightarrow \lambda)^+ (1 \Leftrightarrow x_j)] \quad \leq \quad b + \sum_{j \in L} \lambda x_j + \sum_{j \in N^- \setminus L} z_j \tag{7}$$

is a valid inequality for T given by (6).

5 Polyhedral Results for Formulation A

Consider the MIP model of the SDST flowshop given by (2.1)-(2.8). We are interested in the polyhedral description of the convex hull of the set of feasible solutions. Let $G_{n+1} = (V_{n+1}, A_{n+1})$ be a directed graph on $n+1$ nodes, where each node in the set V_{n+1} is associated with a job in J_0. We assume that G_{n+1} is complete. Thus $|A_{n+1}| = n(n+1)$. Let $X_{n+1} = \{x \in B^{n(n+1)} :$ x is the incidence vector of a tour in $G_{n+1}\}$.

Let $S_A = \{(x, y) \in B^{n(n+1)} \times R^{mn+1} \;:\; (x, y)$ is a feasible solution to (2.2)-(2.8)$\}$, where the y vector includes the mn time variables (2.8) plus the makespan variable $C_{\max}$. Then S_A can be represented as follows: $S_A = \{(x, y) : x \in X_{n+1}, (x, y) \in C_A, y \in Y\}$, where X_{n+1} is the set of constraints involving the binary variables only, $C_A = \{(x, y) \;:\; (x, y)$ satisfies (2.4)$\}$ is the set of coupling constraints involving both binary and real variables, and $Y = \{y \;:\; y$ satisfies (2.5), (2.6), and (2.8)$\}$ is the set of constraints involving the real variables only. It is well known that the set X_{n+1} (the ATSP polytope on $n+1$ nodes) is characterized by (i) assignment constraints and (ii) subtour elimination constraints. In the formulation (2.2)-(2.8), the latter were omitted because they are implied by (2.4) which can be viewed as time-based subtour elimination constraints.

We are interested in the polyhedral structure of $P_A = \text{conv}(S_A)$, the convex hull of S_A. We have $n(n+1)$ binary variables (x_{jk}'s), and $mn+1$ nonnegative real variables (y_{ij}'s and $C_{\max}$) giving a total of $N = n(n+m+1)+1$ variables. Note that once a feasible incidence vector $x \in X_{n+1}$ has been determined, that is, once a given sequence is known, the computation of the associated $y \in R^{mn+1}$ that minimizes the makespan can be done recursively in $O(mn)$ operations.

The following proposition will be used for the main theorem which shows that P_A is full-dimensional.

Proposition 2 *Let θ be a positive real number, $y^0 \in R^t$ be a vector given by $y^0 = \theta(1, 2, \ldots, t \Leftrightarrow 1, t)^T$, and $y^u \in R^t$ be given by $y^u = y^0 + e^u$, where e^u is the u-th unit vector in R^t. Then, the vectors in the set $\{y^0, y^1, y^2, \ldots, y^t\}$ are affinely independent.*

Proof: For $\alpha_0, \alpha_1, \ldots, \alpha_p \in R$, we prove that the following system of linear equations

$$\sum_{u=0}^{t} \alpha_u y^u = 0 \tag{8}$$

$$\sum_{u=0}^{t} \alpha_u = 0 \tag{9}$$

implies $\alpha_u = 0$ for all $u = 0, \ldots, t$.

From (8) we have

$$\begin{aligned} \sum_{u=0}^{t} \alpha_u y^u = 0 \quad &\Rightarrow \quad \alpha_0 y^0 + \sum_{u=1}^{t} \alpha_u (y^0 + e^u) = 0 \\ &\Rightarrow \quad \sum_{u=0}^{t} \alpha_u y^0 + \sum_{u=1}^{t} \alpha_u e^u = 0 \\ &\Rightarrow \quad y^0 \sum_{u=0}^{t} \alpha_u + \sum_{u=1}^{t} \alpha_u e^u = 0 \end{aligned}$$

From (9), we now have

$$\sum_{u=1}^{t} \alpha_u e^u = 0$$

so $\alpha_1 = \alpha_2 = \ldots = \alpha_n = 0$ and hence $\alpha_0 = 0$, which completes the proof. ■

We now state and prove the theorem defining the dimension of P_A.

Theorem 2 *Let $P_A = conv(S_A)$ be the convex hull of S_A. Then* $\dim(P_A) = n(n + m \Leftrightarrow 1)$

Proof: The proof consists of two parts.

(a) It is known that one of the $2(n+1)$ assignment constraints (2.2)-(2.3) is redundant. This implies that $\text{rank}(A^=, b^=) \geq 2n + 1$, where $(A^=, b^=)$ is the equality set of P_A. It follows from Lemma 1 that

$$\begin{aligned} \dim(P_A) &\leq N \Leftrightarrow (2n+1) \\ &= n(n+m+1) + 1 \Leftrightarrow (2n+1) \\ &= n(n+m \Leftrightarrow 1) \end{aligned}$$

(b) To prove $\dim(P_A) \geq n(n + m \Leftrightarrow 1)$ we will show that there exists a set of $n(n + m \Leftrightarrow 1) + 1$ affinely independent vectors in R^N. In this regard, consider the subspace X_{n+1} of P_A. The dimension of the ATSP polyhedron on $n+1$ vertices is $n^2 \Leftrightarrow n \Leftrightarrow 1$ (e.g., see [7]). This implies that there exists a set of $K = n^2 \Leftrightarrow n$ affinely independent vectors $x^1, \ldots, x^K$ in $R^{n(n+1)}$, each being the incident vector of a tour. Also note that for any given $x^t \in X_{n+1}$, there exists a corresponding infinite number of feasible assignments of the time variables for P_A. For each x^t, $t = 2, \ldots, K$, let $y^t \in R^{mn+1}$ be any corresponding feasible assignment of the time variables on P_A. Here, y^t includes the mn time variables y_{ij}, and the makespan variable $C_{\max}$. Hence, the set S_1 given by

$$S_1 = \left\{ \begin{pmatrix} x^2 \\ y^2 \end{pmatrix}, \ldots, \begin{pmatrix} x^K \\ y^K \end{pmatrix} \right\}$$

is a set of feasible (and affinely independent) vectors in R^N, with $|S_1| = K \Leftrightarrow 1 = n^2 \Leftrightarrow n \Leftrightarrow 1$.

For x^1 we construct the corresponding y^1 as follows. Assume for simplicity that x^1 defines the job schedule $(1, 2, \ldots, n)$; that is, $x_{j,j+1} = 1$ for all $j = 0, 1, \ldots, n$ (indices 0 and $n+1$ are the same), and $x_{jk} = 0$ otherwise. Assume also that the $mn + 1$ components of y^1 are given in the order

$$y^1 \quad = \quad \begin{pmatrix} y^1_{11} \\ \vdots \\ y^1_{m1} \\ y^1_{12} \\ \vdots \\ y^1_{m2} \\ \vdots \\ y^1_{1n} \\ y^1_{mn} \\ C_{\max} \end{pmatrix}$$

That is, all the time variables associated with job 1 come first, then those for job 2, and so on, up to job n (the last in the sequence). The makespan variable comes at the end. Now, it is possible to select a large enough number θ such that the following yields a feasible solution for P_A:

$$\begin{pmatrix} y^1_{11} \\ y^1_{21} \\ \vdots \\ y^1_{m1} \\ \vdots \\ y^1_{1n} \\ \vdots \\ y^1_{mn} \\ C_{\max} \end{pmatrix} = \begin{pmatrix} \theta \\ 2\theta \\ \vdots \\ m\theta \\ \vdots \\ ((n-1)m+1)\theta \\ \vdots \\ mn\theta \\ (mn+1)\theta \end{pmatrix}$$

Let e^u be the u-th unit vector in R^{mn+1}, for all $u = 1, \ldots, mn+1$, and denote the vector $y^1 + e^u$ by $y^{1,u}$. By choosing θ as

$$\theta = \max_{ijk} \{p_{ij} + s_{ijk}\} + 1$$

we ensure not only the feasibility of y^1 but the feasibility of $y^{1,u}$ for all $u = 1, \ldots, mn+1$, as well. Using Proposition 2 with $t = mn+1$ and y^1 as the base vector, we conclude that the $mn+2$ vectors in $\{y^1, y^{1,1}, y^{1,2}, \ldots, y^{1,mn+1}\}$ are affinely independent in R^{mn+1}, which in turn implies affine independence in R^N for the points in the set

$$S_2 = \left\{ \begin{pmatrix} x^1 \\ y^1 \end{pmatrix}, \begin{pmatrix} x^1 \\ y^{1,1} \end{pmatrix}, \begin{pmatrix} x^1 \\ y^{1,2} \end{pmatrix}, \ldots, \begin{pmatrix} x^1 \\ y^{1,mn+1} \end{pmatrix} \right\}$$

with $|S_2| = mn+2$.

It is left to show that the vectors in $S_1 \cup S_2$ are affinely independent. Let α_t, β_u be real numbers for $t \in J_1 = \{1, \ldots, K\}$, and $u \in J_2 = \{1, \ldots, mn+1\}$ such that

$$\sum_{t \in J_1} \alpha_t \begin{pmatrix} x^t \\ y^t \end{pmatrix} + \sum_{u \in J_2} \beta_u \begin{pmatrix} x^1 \\ y^{1,u} \end{pmatrix} = 0 \quad (10)$$

$$\sum_{t \in J_1} \alpha_t + \sum_{u \in J_2} \beta_u = 0 \quad (11)$$

This is a linear system of equations for (α_t, β_u). We now prove that this system has a unique zero solution. We distinguish three cases:

Case 1: $\alpha_t = 0$ for all $t \in J_1$

System (10)-(11) reduces to

$$\sum_{u \in J_2} \beta_u \begin{pmatrix} x^1 \\ y^{1,u} \end{pmatrix} = 0$$

$$\sum_{u\in J_2} \beta_u = 0$$

Due to the affine independence of S_2, it follows that $\beta_u = 0$ for $u \in J_2$. Hence, an all-zero solution for (α_t, β_u) is obtained.

Case 2: $\beta_u = 0$ for all $u \in J_2$

The linear system (10)-(11) becomes

$$\begin{aligned}\sum_{t\in J_1} \alpha_t \begin{pmatrix} x^t \\ y^t \end{pmatrix} &= 0 \\ \sum_{t\in J_1} \alpha_t &= 0\end{aligned}$$

which leads to $\alpha_t = 0$ for $t \in J_1$ due to the affine independence of the vectors in S_1.

Case 3: There exists $I_1, I_2 \neq \emptyset$ such that $\alpha_t \neq 0$ for all $t \in I_1 \subseteq J_1$ and $\beta_u \neq 0$ for all $u \in I_2 \subseteq J_2$. Here we have $\alpha_t = 0$ for all $t \in J_1 \setminus I_1$ and $\beta_u = 0$ for all $u \in J_2 \setminus I_2$. We show that Case 3 cannot occur. The corresponding linear system is

$$\begin{aligned}\sum_{t\in I_1} \alpha_t \begin{pmatrix} x^t \\ y^t \end{pmatrix} + \sum_{u\in I_2} \beta_u \begin{pmatrix} x^1 \\ y^{1,u} \end{pmatrix} &= 0 \\ \sum_{t\in I_1} \alpha_t + \sum_{u\in I_2} \beta_u &= 0\end{aligned}$$

which can be rewritten as

$$\sum_{t\in I_1} \alpha_t x^t + x^1 \sum_{u\in I_2} \beta_u = 0 \tag{12}$$

$$\sum_{t\in I_1} \alpha_t y^t + \sum_{u\in I_2} \beta_u y^{1,u} = 0 \tag{13}$$

$$\sum_{t\in I_1} \alpha_t + \sum_{u\in I_2} \beta_u = 0 \tag{14}$$

We first note that $\beta' \equiv \sum_{u\in I_2} \beta_u \neq 0$. Otherwise (12) and (14) would become

$$\begin{aligned}\sum_{t\in I_1} \alpha_t x^t &= 0 \\ \sum_{t\in I_1} \alpha_t &= 0\end{aligned}$$

which implies, due to the affine independence of $\{x^t\}$, that $|I_1| = 0$. This is clearly a contradiction.

Now consider the following two subcases:

Case 3a: $1 \notin I_1$

Equations (12) and (14) become

$$\begin{aligned} \beta' x^1 + \sum_{t \in I_1} \alpha_t x^t &= 0 \\ \beta' + \sum_{t \in I_1} \alpha_t &= 0 \end{aligned}$$

However, this contradicts the affine independence of $\{x^t\}$.

Case 3b: $1 \in I_1$

System (12)-(14) is rewritten as

$$(\alpha_1 + \beta')x^1 + \sum_{t \in I_1 \setminus \{1\}} \alpha_t x^t = 0 \tag{15}$$

$$\sum_{t \in I_1} \alpha_t y^t + \sum_{u \in I_2} \beta_u y^{1,u} = 0 \tag{16}$$

$$(\alpha_1 + \beta') + \sum_{t \in I_1 \setminus \{1\}} \alpha_t = 0 \tag{17}$$

Equations (15) and (17), and the affine independence of $\{x^t\}$ imply that $I_1 \setminus \{1\} = \emptyset$; that is, $I_1 = \{1\}$ consists only of one index. Thus eqs. (16) and (17) become

$$\begin{aligned} \alpha_1 y^1 + \sum_{u \in I_2} \beta_u y^{1,u} &= 0 \\ \alpha_1 + \sum_{u \in I_2} \beta_u &= 0 \end{aligned}$$

which contradicts the affine independence of $\{y^1, y^{1,u}\}$ (by Proposition 2).

This proves that Case 3 cannot occur.

The results from Cases 1 and 2 prove that $S_1 \cup S_2$ is an affine independent set in R^N, the size of set being $n(n+m \Leftrightarrow 1)+1$. We conclude that $\dim(P_A) \geq n(n+m \Leftrightarrow 1)$.

Thus $\dim(P_A) = n(n+m \Leftrightarrow 1)$. ∎

Corollary 2 *The equality set of P_A is given by the assignment constraints (2.2)-(2.3); that is,*

$$(A^{=}, b^{=}) = ((A^{=}_{\mathrm{ATSP}}, O), b^{=})$$

where $A^{=}_{\mathrm{ATSP}}$ is the equality set of the associated ATSP on $n+1$ vertices.

Proof: Lemma 1 and Theorem 2 imply that $\text{rank}(A^{=}, b^{=}) = 2n + 1$, which is the rank of the equality set defined by the assignment constraints. ■

When a proper face F of P_A is found to have dimension $\dim(F) = n(n + m \Leftrightarrow 1) \Leftrightarrow 1$, Theorem 2 implies that F is a facet of P_A. We now establish the following relationship between facets of $\text{conv}(X_{n+1})$ (the ATSP polytope on $n + 1$ nodes) and facets of P_A.

Theorem 3 *Let* $F_{\text{ATSP}} = \{x \in P \, : \, \pi x = \pi_0\}$ *be a facet of conv(*X_{n+1}*). Then*

$$F_A = \{(x, y) \in P_A \, : \, (\pi, 0)(x, y)^T = \pi_0\}$$

is a facet of P_A*.*

Proof: Let F_{ATSP} be a facet of $\text{conv}(X_{n+1})$. Then $\dim(F_{\text{ATSP}}) = \dim(T_{n+1}) \Leftrightarrow 1$, or, expressed in terms of the rank of its equality set,

$$\begin{aligned} \text{rank}\left(\begin{pmatrix} A^{=}_{\text{ATSP}} \\ \pi \end{pmatrix}, \begin{pmatrix} b^{=} \\ \pi_0 \end{pmatrix} \right) &= \text{rank}(A^{=}_{\text{ATSP}}, b^{=}) + 1 \\ &= 2n + 2 \end{aligned}$$

That is, (π, π_0) is linearly independent of the rows of $(A^{=}_{\text{ATSP}}, b^{=})$. Note that $((\pi, 0), \pi_0)$ is a valid inequality for P_A and a nonempty face of P_A. Let $(A^{=}, b^{=})$ be the equality set of P_A. Then $\text{rank}(A^{=}, b^{=}) = 2n + 1$. The equality set of F_A is then given by

$$(A^{=}_{F}, b^{=}_{F}) = \left(\begin{pmatrix} A^{=} \\ \pi' \end{pmatrix}, \begin{pmatrix} b^{=} \\ \pi_0 \end{pmatrix} \right)$$

where $\pi' = (\pi, 0)$. The rank of this equality set either stays the same at $(2n + 1)$ or increases by one to $(2n + 2)$. Assume the former; i.e., that $\text{rank}(A^{=}_{F}, b^{=}_{F}) = 2n + 1$. This would imply that

$$\text{rank}\left(\begin{pmatrix} A^{=}_{\text{ATSP}} & 0 \\ \pi & 0 \end{pmatrix}, \begin{pmatrix} b^{=} \\ \pi_0 \end{pmatrix} \right) = 2n + 1$$

yielding

$$\text{rank}\left(\begin{pmatrix} A^{=}_{\text{ATSP}} \\ \pi \end{pmatrix}, \begin{pmatrix} b^{=} \\ \pi_0 \end{pmatrix} \right) = 2n + 1;$$

which is a contradiction. Therefore, $\text{rank}(A^{=}_{F}, b^{=}_{F}) = 2n + 2$, which gives $\dim(F_A) = \dim(P_A) \Leftrightarrow 1$; i.e., F_A is a facet of P_A. ■

This result is very important in the sense that any known facet of $\text{conv}(X_{n+1})$ can be easily transformed into a facet of P_A by just adding the corresponding zero vector ($0 \in R^{nm+1}$) to the inequality defining the facet in $R^{n(n+1)}$. The identification of such facets would be at the core of any B&C scheme devised to solve the SDST flowshop problem.

5.1 Mixed-Integer Cuts

For the purpose of developing cuts, we rewrite eqs. (2.4) and (2.8) as follows:

$$y_{ij} \Leftrightarrow y_{ik} + (A_i + \tau_{ijk})x_{jk} \quad \leq \quad A_i \quad i \in I,\ j,k \in J \tag{18}$$

$$B_{ij} \quad \leq \quad y_{ij} \quad i \in I,\ j \in J \tag{19}$$

where $\tau_{ijk} = p_{ij} + s_{ijk}$ accounts for both the processing and setup times on machine i. Let $z_{ij} = y_{ij} \Leftrightarrow B_{ij}$, so that $0 \leq z_{ij}$ and define $\xi_{ijk} = (A_i + \tau_{ijk})x_{jk}$. Substituting into (18) gives

$$z_{ij} \Leftrightarrow z_{ik} + \xi_{ijk} \quad \leq \quad A_i \Leftrightarrow B_{ij} + B_{ik} \tag{20}$$

Now, we apply Proposition 1 with $N^+ = \{ij, ijk\}$, $N^- = \{ik\}$, $C = \{ij\}$, and $L = \emptyset$. If C is a dependent set; that is, if $\lambda = \tau_{ijk} + B_{ij} \Leftrightarrow B_{ik} > 0$, then (20) gives rise to the valid inequality

$$\xi_{ijk} + (A_i \Leftrightarrow B_{ij} + B_{ik})^+(1 \Leftrightarrow x_{jk}) \quad \leq \quad A_i \Leftrightarrow B_{ij} + B_{ik} + z_{ik} \tag{21}$$

Assuming $(A_i \Leftrightarrow B_{ij} + B_{ik})^+ > 0$, (21) becomes

$$\xi_{ijk} \Leftrightarrow (A_i \Leftrightarrow B_{ij} + B_{ik})x_{jk} \quad \leq \quad z_{ik} \qquad \text{or}$$

$$(p_{ij} + s_{ijk} + B_{ij} \Leftrightarrow B_{ik})x_{jk} \Leftrightarrow y_{ik} \quad \leq \quad \Leftrightarrow B_{ik} \tag{22}$$

which is the desired result. Inequality (22) will have an effect only if $(p_{ij} + s_{ijk} + B_{ij} \Leftrightarrow B_{ik}) > 0$; that is, if C, as chosen, is a dependent set. Note that when $x_{jk} = 1$, (22) becomes $B_{ij} + p_{ij} + s_{ijk} \leq y_{ik}$ as expected and when $x_{jk} = 0$, it reduces to $B_{ik} \leq y_{ik}$, the default bound.

6 Polyhedral Results for Formulation B

Now consider the MIP model of the SDST flowshop given by (3.1)-(3.7). Let $S = \{S_i\}$, for $i = 1, 2, \ldots, n!$, be the set of all feasible schedules. For every schedule $S_i \in S$ there exists an incidence vector $x^i \in B^{n(n-1)/2}$. Let $\tilde{X}_n = \{x \in B^{n(n-1)/2} : x \text{ is the incidence vector of a schedule}\}$.

Paralleling the notation in the previous section, let

$$S_B = \{(x, y) \in B^{n(n-1)/2} \times R^{mn+1} : (x, y) \text{ is a feasible solution to (3.2)-(3.7)}\}$$

Again, the y vector includes the mn time variables (3.7) plus the makespan variable $C_{\max}$. The set S_B can be represented as follows: $S_B = \{(x, y) : x \in \hat{X}_n, (x, y) \in C_B, y \in Y\}$, where $\hat{X}_n$ is the set of constraints involving the binary variables only, $C_B = \{(x, y) : (x, y) \text{ satisfies (3.2)-(3.3)}\}$ is the set of coupling constraints involving both binary and real variables, and $Y = \{y : y \text{ satisfies (3.4), (3.5), and (3.7)}\}$ is the set of constraints involving the real variables only. Note that this set Y is the same as defined in the previous section.

We are interested in the polyhedral structure of $P_B = \text{conv}(S_B)$, the convex hull of S_B. Of particular interest is $\text{conv}(\hat{X}_n)$, the convex hull of $\hat{X}_n$ and its relationship to P_B. In contrast with formulation A, and the related polytope $\text{conv}(X_{n+1})$, the corresponding subspace $\hat{X}_n$ in formulation B has yet to be unexplored. In this section we first provide a more detailed study of the scheduling polyhedron $\text{conv}(\hat{X}_n)$. Subsequently, we give some results that link $\text{conv}(\hat{X}_n)$ with P_B, which in a sense, parallel those that allowed us to extend the polyhedral structure of $\text{conv}(X_{n+1})$ to P_A in the previous section.

6.1 The $\text{conv}(\hat{X}_n)$ Polyhedron

Throughout this section, we assume that the components of a feasible $x \in \hat{X}_n$ are stored columnwise; i.e., in the following order:

$$x = (x_{12}, x_{13}, x_{23}, \ldots, x_{1,n-1}, x_{2,n-1}, \ldots, x_{n-2,n-1}, x_{1,n}, x_{2,n}, \ldots, x_{n-1,n})^T$$

so $x \in B^{n(n-1)/2}$.

Lemma 2 *Conv($\hat{X}_n$) is full-dimensional; i.e.,* $\dim(\hat{X}_n) = \frac{n(n-1)}{2}$.

Proof: By induction on n. For $n = 2$ there are only two schedules, $S_1 = (1, 2)$ and $S_2 = (2, 1)$, with corresponding incidence one-dimensional vectors $x^1 = (1)$ and $x^2 = (0)$, respectively. Hence, $\text{conv}(\hat{X}_2)$ is given by $\text{conv}(\hat{X}_2) = \{x \in R : 0 \leq x \leq 1\}$. Clearly, $x = 1/2$ is an interior point of $\text{conv}(\hat{X}_2)$. It follows from Corollary 1 that $\hat{X}_2$ is full-dimensional.

Now assume the induction hypothesis; that is, that $\text{conv}(\hat{X}_n)$ is full-dimensional. By implication there exists a set of $N + 1$ affinely independent points $\{x^1, \ldots, x^N, x^{N+1}\}$, where $N = \dim(\hat{X}_n) = \frac{n(n-1)}{2}$ and each $x^i \in \hat{X}_n$ in the set is the incidence vector of a schedule. We need to prove that $\text{conv}(\hat{X}_{n+1})$ is full-dimensional.

In $\hat{X}_{n+1}$ there is an extra job to be scheduled (job $n+1$). The corresponding points have n additional coordinates with respect to the points in $\hat{X}_n$ given by the variables $x_{1,n+1}, x_{2,n+1}, \ldots, x_{n,n+1}$. Note that for any $x^i \in \hat{X}_n$, the assignment $x^i_{1,n+1} = x^i_{2,n+1} = \ldots = x^i_{n,n+1} = 0$ (which correspond to scheduling job $n + 1$ at the beginning of S_i) yields a feasible schedule for $\hat{X}_{n+1}$ so

$$\left\{ \begin{pmatrix} x^1 \\ 0 \end{pmatrix}, \begin{pmatrix} x^2 \\ 0 \end{pmatrix}, \ldots, \begin{pmatrix} x^{N+1} \\ 0 \end{pmatrix} \right\} \subseteq \hat{X}_{n+1} \subseteq \text{conv}(\hat{X}_{n+1}).$$

Moreover, these vectors are affinely independent.

For a given $x^i \in X_n$, say x^1, we build n vectors in X_{n+1} as follows. Taking $x^1 \in X_n$ as a common base, we append the n-dimensional vector $v^j = (x^j_{1,n+1}, \ldots, x^j_{n,n+1})^T$, such that $(x^1, v^j)^T \in \hat{X}_{n+1}$, for $j = 1, \ldots, n$. Here, the components of v^j are determined when job $n + 1$ is scheduled right after the j-th scheduled job in S_1, for $j = 1, \ldots, n$. For instance, assuming for simplicity that x^1 is the incidence vector of $S_1 = (1, 2, \ldots, n)$, then

$$\begin{array}{llll} \text{Insert } n+1 \text{ after } 1 & \Rightarrow & (1, n+1, 2, \ldots, n) & \Rightarrow \quad v^1 = (1, 0, \ldots, 0) \\ \text{Insert } n+1 \text{ after } 2 & \Rightarrow & (1, 2, n+1, 3, \ldots, n) & \Rightarrow \quad v^2 = (1, 1, 0, \ldots, 0) \\ \quad\vdots & & & \\ \text{Insert } n+1 \text{ after } n & \Rightarrow & (1, \ldots, n, n+1) & \Rightarrow \quad v^n = (1, 1, \ldots, 1). \end{array}$$

Note that the vectors in $\{v^j\}$ are linearly independent. The set

$$\left\{ \begin{pmatrix} x^1 \\ 0 \end{pmatrix}, \ldots, \begin{pmatrix} x^{N+1} \\ 0 \end{pmatrix}, \begin{pmatrix} x^1 \\ v^1 \end{pmatrix}, \ldots, \begin{pmatrix} x^1 \\ v^n \end{pmatrix} \right\}$$

has dimension $N + 1 + n = n(n \Leftrightarrow 1)/2 + 1 + n = (n+1)n/2 + 1 = \dim(\hat{X}_{n+1}) + 1$. It remains to prove that these $N + 1 + n$ vectors are affinely independent. To do so, consider the following system of linear equations in (α_i, β_j), for $i = 1, \ldots, N+1$, $j = 1, \ldots, n$:

$$\begin{aligned} \sum_i \alpha_i \begin{pmatrix} x^i \\ 0 \end{pmatrix} + \sum_j \beta_j \begin{pmatrix} x^1 \\ v^j \end{pmatrix} &= 0 \\ \sum_i \alpha_i + \sum_j \beta_j &= 0 \end{aligned}$$

This system can be rewritten as

$$\sum_i \alpha_i x^i + \sum_j \beta_j x^1 = 0 \tag{23}$$

$$\sum_j \beta_j v^j = 0 \tag{24}$$

$$\sum_i \alpha_i + \sum_j \beta_j = 0. \tag{25}$$

Equation (24) and the fact that $\{v^j\}$ are linearly independent imply $\beta_j = 0$ for all j. Thus (23)-(25) reduces to

$$\begin{aligned} \sum_i \alpha_i x^i &= 0 \\ \sum_i \alpha_i &= 0 \end{aligned}$$

It follows from the affine independence of $\{x^i\}$ that $\alpha_i = 0$ for all i. Therefore, the $(n+1)n/2 + 1$ vectors are affinely independent so $\dim(\text{conv}(\hat{X}_{n+1})) = (n+1)n/2$ implying that $\text{conv}(\hat{X}_{n+1})$ is full-dimensional. ■

6.2 Facets of conv($\hat{X}_n$)

We observe that $\hat{X}_n$ has certain symmetry in the sense that if $x \in \hat{X}_n$ then $\bar{x} \in \hat{X}_n$, where $\bar{x} = (1 \Leftrightarrow x_{12}, \ldots, 1 \Leftrightarrow x_{1n}, \ldots, 1 \Leftrightarrow x_{n-1,n})$ is the componentwise complement of x. This leads to the following lemma.

Lemma 3 *$F = \{x \in conv(\hat{X}_n) : \pi x = \pi_0\}$ is a facet of conv($\hat{X}_n$) if and only if $\bar{F} = \{x \in conv(\hat{X}_n) : \Leftrightarrow \pi x = \pi_0 \Leftrightarrow \sum_{jk} \pi_{jk}\}$ is a facet of conv($\hat{X}_n$), where $\sum_{jk} \pi_{jk}$ is the sum of all components of vector π.*

Proof: Since F is a facet of conv($\hat{X}_n$), $\dim(F) = \dim(\text{conv}(\hat{X}_n)) \Leftrightarrow 1$ (by Lemma 1). Hence, there exists $K = \dim(\text{conv}(\hat{X}_n))$ affine independent vectors $x^i \in F$. Consider the vectors $\{\bar{x}^i\}$. It is easy to verify that $\bar{x}^i \in \bar{F}$. Furthermore, all the $\bar{x}^i$ are affinely independent as well, as shown below.

$$\begin{aligned}
\sum_i \alpha_i \bar{x}^i = 0 \quad \text{and} \quad \sum_i \alpha_i = 0 \quad &\Longrightarrow \quad \sum_i \alpha_i (\mathbf{1} \Leftrightarrow x^i) = 0 \quad \text{and} \quad \sum_i \alpha_i = 0 \\
&\Longrightarrow \quad \mathbf{1} \sum_i \alpha_i \Leftrightarrow \sum_i \alpha_i x^i = 0 \quad \text{and} \quad \sum_i \alpha_i = 0 \\
&\Longrightarrow \quad \sum_i \alpha_i x^i = 0 \quad \text{and} \quad \sum_i \alpha_i = 0 \\
&\Longrightarrow \quad \alpha_i = 0 \quad \text{for all } i
\end{aligned}$$

due to the affine independence of the x^i vectors, where $\mathbf{1}$ is a vector with each component equal to 1. It follows that $\dim(\bar{F}) = K \Leftrightarrow 1$ and that $\bar{F}$ is a facet of conv($\hat{X}_n$). The converse is shown similarly. ■

Basically, Lemma 3 establishes that for every facet of conv($\hat{X}_n$) there is a symmetric counterpart which is also a facet of conv($\hat{X}_n$) and tells us how to find it.

Proposition 3 *The nonnegativity constraints*

$$x_{jk} \geq 0 \qquad j, k \in J, j < k$$

give facets of conv(X_n) for $n \geq 2$.

Proof: Let $j, k \in J, j < k$. Let $\pi x \leq 0$ represent the constraint $\Leftrightarrow x_{jk} \leq 0$, that is, $\pi = (0, \ldots, 0, \Leftrightarrow 1, 0, \ldots, 0)$ and $\pi_0 = 0$ where the -1 component in π corresponds to π_{jk}. Note that

(a) $\pi x \leq \pi_0$ is a valid inequality of conv($\hat{X}_n$), so $F = \{x \in \text{conv}(\hat{X}_n) : \pi x = \pi_0\}$ is a face of conv(X_n).

(b) F is a proper face since $\pi x \leq \pi_0$ is satisfied at equality by some $x^i \in \hat{X}_n$ and is a strict inequality for some other $x^i \in \hat{X}_n$. In fact, any schedule S_i where job j is after (before) job k satisfies $\pi x \leq 0$ as an equality (strict inequality).

We prove the result by showing that conditions of Theorem 1 hold. Here $\pi x \leq \pi_0$ represents a nonnegativity constraint, the equality set $(A^=, b^=)$ does not exist since conv(X_n) is full-dimensional, and we are concerned with solutions to the linear system

$$\lambda x^i = \lambda_0, \tag{26}$$

where x^i is the incidence vector of schedule S_i (with components stored row-wise) and $\{S_i\}$ is the set of schedules that satisfy $\pi x^i = \pi_0$. Hence, it suffices to demonstrate that all solutions (λ, λ_0) to (26) for all i are of the form $\lambda = \alpha\pi$, $\lambda_0 = \alpha\pi_0$ for some $\alpha \in R$.

Because $\{S_i\}$ is the set of schedules satisfying $\pi x^i = \pi_0$, that is $x^i_{jk} = 0$, then $\{S_i\}$ contains all schedules where job k is scheduled before job j. In particular, $S_0 = (n, n \Leftrightarrow 1, \ldots, k, \ldots, j, \ldots, 2, 1) \in \{S_i\}$ and $x^0 = 0 \in B^{\frac{n(n-1)}{2}}$. Thus

$$\lambda x^0 = \lambda_0 \Leftrightarrow \lambda \cdot 0 = \lambda_0 \Leftrightarrow \lambda_0 = 0$$

so system (26) reduces to $\lambda x = 0$. To determine the solution $\lambda = (\lambda_{12}, \lambda_{13}, \ldots, \lambda_{1n}, \ldots, \lambda_{n-1,n}) \in R^{\frac{n(n-1)}{2}}$ we proceed as follows. From S_0 we obtain S_1 by swapping jobs 2 and 1 such that

$$S_1 = (n, n \Leftrightarrow 1, \ldots, k, \ldots, j, \ldots, 3, 1, 2) \in \{S_i\}$$

with corresponding incidence vector $x^1 = (1, 0, \ldots, 0)$. Thus

$$\lambda x^1 = 0 \Leftrightarrow \lambda_{12} = 0.$$

Similarly, we obtain S_2 by swapping jobs 3 and 1:

$$S_2 = (n, n \Leftrightarrow 1, \ldots, k, \ldots, j, \ldots, 4, 1, 3, 2) \in \{S_i\}$$

with $x^2 = (1, 1, 0, \ldots, 0)$. Thus

$$\lambda x^1 = 0 \Leftrightarrow \lambda_{13} = 0$$

because we already have found that $\lambda_{12} = 0$.

Observe that every time we swap two adjacent jobs u, v, the corresponding incidence vectors are equal except for the component associated with these jobs x_{uv}. Also, as long as jobs j and k are not swapped, the resulting schedule remains feasible and satisfies $\pi x = \pi_0$. Therefore, by swapping job 1 with jobs $4, 5, \ldots, n$ (one at a time), we arrive at the schedule $S_{n-1} = (1, n, n \Leftrightarrow 1, \ldots, k, \ldots, j, \ldots, 3, 2)$, finding along the way that $\lambda_{14} = \ldots = \lambda_{1n} = 0$; that is, $\lambda_{1q} = 0$ for all $q = 2, \ldots, n$.

Proceeding similarly with jobs $2, 3, \ldots, j \Leftrightarrow 1$, and evaluating (26) for each generated x^i, we find $\lambda_{pq} = 0$ for all $p = 1, \ldots, j \Leftrightarrow 1$ and $q = p + 1, \ldots, n$. After the final swap, we have

$$S_l = (1, 2, \ldots, j \Leftrightarrow 2, j \Leftrightarrow 1, n, n \Leftrightarrow 1, \ldots, k, \ldots, j)$$

for some l.

By shifting one at a time the jobs in S_0 scheduled after job j, and by substituting the corresponding x^i in system (26), we have recursively found that $\lambda_{pq} = 0$ for all p, q such that $p < j$. If instead of shifting the jobs at the end of the schedule (after job j), we carry out the same procedure starting with the jobs at the beginning of the schedule (before job k) we arrive at the conclusion that $\lambda_{pq} = 0$ for all p, q such that $q > k$. That is, given S_l, swap jobs n and $n - 1$ to get

$$S_{l+1} = (1, \ldots, j - 1, n - 1, n, n - 2, \ldots, k, \ldots, j)$$

Then, $\lambda x^{l+1} = 0$ implies $\lambda_{n-1,n} = 0$. Keep on swapping job n with each of the jobs $n - 2, n - 3, \ldots, j$ one at a time to obtain $\lambda_{n-2,n} = \lambda_{n-3,n} = \ldots = \lambda_{j+1,n}, \lambda_{j,n} = 0$. After the last exchange, we have the schedule $S_{l+n-j} = (1, \ldots, j - 1, n - 1, n - 2, \ldots, k, \ldots, j, n)$. Repeat recursively this shifting procedure for jobs $n - 1, n - 2, \ldots, k + 1$, to obtain $\lambda_{pq} = 0$ for all p, q such that $q = n, n - 1, \ldots, k + 1$, with final schedule $S_r = (1, \ldots, j - 1, k, k - 1, \ldots, j + 1, j, k + 1, k + 2, \ldots, n - 1, n)$, for some r.

It remains to determine λ_{pq} for all p, q such that $p = j, j + 1, \ldots, k - 1$ and $q = p + 1, \ldots, k$. However, by applying the same reasoning, we swap job k and $k - 1$ to get $\lambda_{k-1,k} = 0$. Then we swap job k with $k - 2$ and so on up to job $j + 1$. This leads to $\lambda_{k-2,k} = \lambda_{k-3,k} = \ldots, \lambda_{j+1,k} = 0$ with the corresponding schedule $S_{r+k-j} = (\ldots, k - 1, k - 2, k - 3, \ldots, j + 1, k, j, k - 1, \ldots)$. Repeating these operations for job $k - 2, k - 3, \ldots, j + 1$, but shifting all the way up to job j, we find $\lambda_{pq} = 0$ for all remaining (p, q) pairs except (j, k). The resulting schedule is $S_s = (1, \ldots, j - 1, k, j, j + 1, \ldots, k - 1, k + 1, \ldots, n)$, for some s. Therefore, $\lambda_{pq} = 0$ for all $(p, q) \neq (j, k)$.

Hence, a solution for (26) is given by $(\lambda, 0)$, where $\lambda = (0, \ldots, 0, \lambda_{jk}, 0, \ldots, 0)$. It is straightforward to check that $\alpha = -\lambda_{jk}$ satisfies

$$\lambda = \alpha \pi \quad \text{and} \quad \lambda_0 = \alpha \pi_0$$

as was to be shown. ■

Corollary 3 *The inequalities*

$$x_{jk} \leq 1 \quad j, k \in J, j < k$$

give facets of conv(X_n) for all $n \geq 2$.

Proof: Follows from Proposition 3 and Lemma 3. ■

In contrast with X_{n+1} in model A, it is not possible to identify analogous ATSP valid inequalities such as subtour elimination constraints, comb inequalities, and D_k^+, D_k^- inequalities for model B. One set of valid inequalities that we can identify, though, corresponds to precedence violations for a sequence of jobs. Table 3 shows the valid inequalities that eliminate "cycles" (in the precedence sense) for any 3-job subsequence. We call these inequalities, for a subsequence of size t, the t-subsequence elimination constraint (or t-SEC). For $t = 3$ we show below that the 3-SEC are facets of conv($\hat{X}_n$).

Lemma 4 *The inequalities (3-subsequence elimination constraints)*

$$x_{jk} - x_{jl} + x_{kl} \quad \geq \quad 0 \quad j, k, l \in J, \ j < k < l \tag{27}$$

give facets of conv($\hat{X}_n$) for all $n \geq 2$.

sequence	constraint		
$j \rightarrow k \rightarrow l \Rightarrow j \rightarrow l$	$x_{jk} + x_{kl}$	$\leq$	$1 + x_{jl}$
$j \rightarrow l \rightarrow k \Rightarrow j \rightarrow k$	$x_{jl} + (1 \Leftrightarrow x_{kl})$	$\leq$	$1 + x_{jk}$

Table 3: *3-SECs for conv($\hat{X}_n$)*

Proof: Each inequality in (27) represents a proper face of conv($\hat{X}_n$) since it is satisfied as an equality by some schedule (e.g., $S = (l, k, j, \ldots)$) and as a strict inequality for some other schedule (e.g., $S = (l, j, k, \ldots)$).

Again we prove the result by showing the conditions of Theorem 1. Here, $\pi x \leq \pi_0$ is given by $\pi = (0, \ldots, 0, \pi_{jk}, 0, \ldots, 0, \pi_{jl}, 0, \ldots, 0, \pi_{kl}, 0, \ldots, 0)$ and $\pi_0 = 0$, where $\pi_{jk} = \pi_{jl} = \Leftrightarrow 1$ and $\pi_{kl} = 1$. Note that because conv($\hat{X}_n$) is full-dimensional, there is no equality set $(A^=, b^=)$.

Let $\{S_i\}$ be the set of schedules that satisfy $\pi x^i = \pi_0$, for all i. We are concerned with solutions to the linear system

$$\lambda x^i \quad = \quad \lambda_0 \tag{28}$$

where x^i is the incidence vector corresponding to schedule S_i. It suffices to demonstrate that all solutions (λ, λ_0) to (28) are of the form $\lambda = \alpha\pi, \lambda_0 = \alpha\pi_0$ for some $\alpha \in R$.

Equation $\pi x = \pi_0$ (that is, $x_{jk} \Leftrightarrow x_{jl} + x_{kl} = 0$) is satisfied if one of the following three cases occur:

(i) $x_{jk} = x_{jl} = x_{kl} = 0$, which corresponds to $S_i = (\ldots, l, \ldots, k, \ldots, j, \ldots)$.

(ii) $x_{jk} = 0, x_{jl} = x_{kl} = 1$ which corresponds to $S_i = (\ldots, k, \ldots, j, \ldots, l, \ldots)$.

(iii) $x_{jk} = x_{jl} = 1, x_{kl} = 0$, which corresponds to $S_i = (\ldots, j, \ldots, l, \ldots, k, \ldots)$.

Since $S_0 = (n, n \Leftrightarrow 1, \ldots, 2, 1) \in \{S_i\}$ (case (i)), then

$$\lambda x^0 = \lambda_0 \Leftrightarrow \lambda \cdot 0 = \lambda_0 \Leftrightarrow \lambda_0 = 0.$$

By performing the same job shifting procedure we used in the proof of Proposition 3 for the schedules associated with case (i), we find $\lambda_{pq} = 0$ for all $(p, q) \notin \{(j, k), (j, l), (k, l)\}$. Thus, (28) becomes

$$\lambda_{jk} x_{jk} + \lambda_{jl} x_{jl} + \lambda_{kl} x_{kl} = 0.$$

Case (ii) and (iii) imply

$$\begin{aligned} \lambda_{jl} + \lambda_{kl} &= 0 \\ \lambda_{jk} + \lambda_{jl} &= 0 \end{aligned}$$

which is a 2×3 system with solution $\lambda_{jl} = \beta, \lambda_{jk} = \lambda_{kl} = \Leftrightarrow\beta$ for any $\beta \in R$. Hence, by taking $\alpha = \beta$, (λ, λ_0) is given by $(\lambda, \lambda_0) = (\alpha\pi, \alpha\pi_0)$. This completes the proof. ∎

Lemma 5 *The inequalities*

$$x_{jk} \Leftrightarrow x_{jl} + x_{kl} \leq 1 \quad j, k, l \in J,\ j < k < l$$

give facets of conv($\hat{X}_n$) for all $n \geq 2$.

Proof: Follows from Lemma 4 and Lemma 3. ∎

All 4-SECs are shown in Table 4 for all $j, k, l, m \in J,\ j < k < l < m$. These valid inequalities, however, do not define facets of conv($\hat{X}_n$). In fact, because $\dim(\hat{X}_n) = n(n \Leftrightarrow 1)2$ and each 4-SEC can be expressed as the intersection of two of the previously developed facets of conv($\hat{X}_n$) (i.e., combinations of $x_{jk} \geq 0, x_{jk} \leq 1$, and 3-SEC), they define faces of dimension $n(n \Leftrightarrow 1)/2 \Leftrightarrow 2$.

sequence	constraint		
$j \rightarrow k \rightarrow l \rightarrow m \Rightarrow j \rightarrow m$	$x_{jk} + x_{kl} + x_{lm}$	$\leq$	$2 + x_{jm}$
$j \rightarrow k \rightarrow m \rightarrow l \Rightarrow j \rightarrow l$	$x_{jk} + x_{km} + (1 \Leftrightarrow x_{lm})$	$\leq$	$2 + x_{jl}$
$j \rightarrow l \rightarrow k \rightarrow m \Rightarrow j \rightarrow m$	$x_{jl} + (1 \Leftrightarrow x_{kl}) + x_{km}$	$\leq$	$2 + x_{jm}$
$j \rightarrow l \rightarrow m \rightarrow k \Rightarrow j \rightarrow k$	$x_{jl} + x_{lm} + (1 \Leftrightarrow x_{km})$	$\leq$	$2 + x_{jk}$
$j \rightarrow m \rightarrow k \rightarrow l \Rightarrow j \rightarrow l$	$x_{jm} + (1 \Leftrightarrow x_{km}) + x_{kl}$	$\leq$	$2 + x_{jl}$
$j \rightarrow m \rightarrow l \rightarrow k \Rightarrow j \rightarrow k$	$x_{jm} + (1 \Leftrightarrow x_{lm}) + (1 \Leftrightarrow x_{kl})$	$\leq$	$2 + x_{jk}$

Table 4: *4-SECs for conv($\hat{X}_n$)*

The conv($\hat{X}_n$) polytope can be used to model other scheduling problems, such as single-machine and permutation flowshops problems, where every schedule is feasible. When real variables are introduced in the scheduling model, it remains to be determined whether the valid inequalities discussed above define facets of the complete polyhedron. In the next section we prove that this is the case for the SDST flowshop polyhedron.

6.3 The P_B Polyhedron

We now state and prove the theorem defining the dimension of P_B. The proof is very similar to the proof of Theorem 2 because a point $x \in \hat{X}_n$ defines a given feasible sequence for P_B just as $x \in X_{n+1}$ defines a feasible sequence for P_A; moreover, the definition of $y \in R^{mn+1}$ is the same for both polyhedrons.

Theorem 4 *Let P_B = conv(S_B) be the convex hull of S_B. Then P_B is full-dimensional; i.e.,* $\dim(P_B) = n(n \Leftrightarrow 1)/2 + mn + 1$

Proof: Let $N = n(n-1)/2 + mn + 1$. We will show that there exists a set of $N+1$ affinely independent vectors in R^N.

Consider the subspace $\hat{X}_n$ of P_B. We proved in Lemma 2 that conv$(\hat{X}_n)$ is full-dimensional. This implies that there exists a set of $K = n(n-1)/2 + 1$ affinely independent vectors $x^1, \ldots, x^K$ in $R^{n(n+1)}$, each being the incidence vector of a schedule. Also note that for any given $x^t \in \hat{X}_n$, there exists a corresponding infinite number of feasible assignments of the time variables for P_B.

From this point on the rest of the proof follows that of Theorem 2, part (b). We will just sketch the arguments. From the set $\{x^1, \ldots, x^K\}$ we build two disjoint sets $S_1, S_2 \subseteq R^N$ given by

$$S_1 = \left\{ \begin{pmatrix} x^2 \\ y^2 \end{pmatrix}, \ldots, \begin{pmatrix} x^K \\ y^K \end{pmatrix} \right\}$$

$$S_2 = \left\{ \begin{pmatrix} x^1 \\ y^1 \end{pmatrix}, \begin{pmatrix} x^1 \\ y^{1,1} \end{pmatrix}, \begin{pmatrix} x^1 \\ y^{1,2} \end{pmatrix}, \ldots, \begin{pmatrix} x^1 \\ y^{1,mn+1} \end{pmatrix} \right\}$$

where S_1 and S_2 are sets of feasible (and affinely independent) vectors in R^N, with $|S_1| = K - 1 = n(n-1)/2$ and $|S_2| = mn + 2$, so that $|S_1 \cup S_2| = n(n-1)/2 + mn + 2$. We then can prove that the points in $S_1 \cup S_2$ are affinely independent by showing that the linear system

$$\sum_{t \in J_1} \alpha_t \begin{pmatrix} x^t \\ y^t \end{pmatrix} + \sum_{u \in J_2} \beta_u \begin{pmatrix} x^1 \\ y^{1,u} \end{pmatrix} = 0$$

$$\sum_{t \in J_1} \alpha_t + \sum_{u \in J_2} \beta_u = 0$$

admits the unique solution $\alpha_t = \beta_u = 0$ for $t \in J_1 = \{1, \ldots, K\}$, and $u \in J_2 = \{1, \ldots, mn+1\}$. This leads to conclude that $\dim(P_B) = n(n-1)/2 + mn + 1$. ∎

We now establish the following relationship between facets of conv$(\hat{X}_n)$ and facets of P_B.

Theorem 5 *Let $F_X = \{x \in conv(\hat{X}_n) : \pi x = \pi_0\}$ be a facet of conv$(\hat{X}_n)$. Then $F_B = \{(x,y) \in P_B : (\pi, 0)(x,y)^T = \pi_0\}$ is a facet of P_B.*

Proof: Let F_X be a facet of conv$(\hat{X}_n)$. Let (π', π_0) represent the inequality $\pi' z \leq \pi_0$ where $\pi' = (\pi, 0) \in R^N$ and $z = (x, y) \in P_B$. Hence F_B can be rewritten as $F_B = \{z \in P_B : \pi' z = \pi_0\}$. Given that F_X is a facet of conv$(\hat{X}_n)$, it follows that F_B is a proper face of P_B.

We prove the result by showing that conditions of Theorem 1 hold. Here, the equality set $(A^=, b^=)$ does not exist since P_B is full-dimensional, and we are concerned with solutions to the linear system

$$\lambda z = \lambda_0 \tag{29}$$

where z is any point in P_B satisfying $\pi' z = \pi_0$. Hence, it suffices to demonstrate that all solutions (λ, λ_0) to (29) are of the form $\lambda = \alpha\pi, \lambda_0 = \alpha\pi_0$ for some $\alpha \in R$.

Since $z = (x, y) \in P_B$, the system in (29) can be rewritten as

$$\lambda_x x + \lambda_y y \quad = \quad \lambda_0. \tag{30}$$

Let $x^1 \in F_X$. According to the procedure described in the proof of Theorem 4, it is possible to construct $mn + 2$ feasible affinely independent points $y^0, y^1, \ldots, y^{mn+1}$, where $y^u = y^0 + e^u$ for all $u = 1, \ldots, mn + 1$. Here e^u denotes the u-th unit vector in R^{mn+1}. It easy to see that $z^i = (x^1, y^i) \in P_B$ for all $i = 0, \ldots, mn + 1$. Moreover, z^i satisfies $\pi' z^i = \pi x^1 = \pi_0$ for all i so that $z^i \in F_B$. Substituting these $mn + 2$ points in system (29) we have

$$\begin{aligned}
\lambda_x x^1 + \lambda_y y^0 \quad &= \quad \lambda_0 && (31)\\
\lambda_x x^1 + \lambda_y y^1 \quad &= \quad \lambda_0 && (32)\\
&\vdots \\
\lambda_x x^1 + \lambda_y y^{mn+1} \quad &= \quad \lambda_0 && (33)
\end{aligned}$$

By subtracting (31) from all other eqs. (32)-(33), we obtain the following system of order $mn + 1$:

$$\begin{aligned}
\lambda_y(y^1 \Leftrightarrow y^0) \quad &= \quad 0 \\
&\vdots \\
\lambda_y(y^{mn+1} \Leftrightarrow y^0) \quad &= \quad 0
\end{aligned}$$

Since $y^i \Leftrightarrow y^0 = e^i$ it follows that $\lambda_y = 0 \in R^{mn+1}$. This reduces (29) to

$$\lambda_x x = \lambda_0$$

where x satisfies $\pi x = \pi_0$. Given that F_X is a facet, it follows that there is $\alpha \in R$ such that $\lambda_x = \alpha\pi, \lambda_0 = \alpha\pi_0$. This implies that $\lambda = (\lambda_x, \lambda_y) = (\alpha\pi, \alpha 0) = \alpha(\pi, 0) = \alpha\pi'$ and the proof is complete. ■

6.4 Mixed-Integer Cuts

Note that inequalities (3.2) and (3.7) in model B have the same structure as inequalities (2.4) and (2.8) in model A. Thus the valid inequality derived from these equations for model A also applies for model B; that is,

$$(p_{ij} + s_{ijk} + B_{ij} \Leftrightarrow B_{ik})x_{jk} \Leftrightarrow y_{ik} \quad \leq \quad \Leftrightarrow B_{ik} \tag{34}$$

is a valid inequality for model B. Recall that (34) will have an effect only if $(p_{ij} + s_{ijk} + B_{ij} \Leftrightarrow B_{ik}) > 0$. Note that when $x_{jk} = 1$, (34) becomes $B_{ij} + p_{ij} + s_{ijk} \leq y_{ik}$ as expected and when $x_{jk} = 0$, it reduces to $B_{ik} \leq y_{ik}$, the default bound.

In a similar fashion, we use inequalities (3.3) and (3.7), a change of variable $x'_{jk} = 1 \Leftrightarrow x_{jk}$ in (3.3), and the same procedure to derive the valid inequality

$$(p_{ik} + s_{ikj} + B_{ik} \Leftrightarrow B_{ij})(1 \Leftrightarrow x_{jk}) \Leftrightarrow y_{ij} \quad \leq \quad \Leftrightarrow B_{ij}$$

for model B, where again we must have $(p_{ik} + s_{ikj} + B_{ik} \Leftrightarrow B_{ij}) > 0$ for the inequality to be useful.

References

[1] E. Balas. The asymmetric assignment problem and some new facets of the traveling salesman polytope on a directed graph. *SIAM Journal on Discrete Mathematics*, 2(4):425–451, 1989.

[2] E. Balas and M. Fischetti. The fixed-outdegree 1-arborescence polytope. *Mathematics of Operations Research*, 17(4):1001–1018, 1992.

[3] E. Balas and M. Fischetti. A lifting procedure for the asymmetric traveling salesman polytope and a large new class of facets. *Mathematical Programming*, 58(3):325–352, 1993.

[4] B. D. Corwin and A. O. Esogbue. Two machine flow shop scheduling problems with sequence dependent setup times: A dynamic programming approach. *Naval Research Logistics Quarterly*, 21(3):515–524, 1974.

[5] S. E. Dreyfus and A. M. Law. *The Art and Theory of Dynamic Programming.* Academic Press, Orlando, 1977.

[6] M. Fischetti. Facets of the asymmetric traveling salesman polytope. *Mathematics of Operations Research*, 16(1):42–56, 1991.

[7] N. Grötschell and M. W. Padberg. Polyhedral theory. In E. L. Lawler, J. K. Lenstra, A. H. G. Rinnoy Kan, and D. B. Shmoys, editors, *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*, pages 251–305. John Wiley & Sons, Chichester, 1985.

[8] J. N. D. Gupta and W. P. Darrow. The two-machine sequence dependent flowshop scheduling problem. *European Journal of Operational Research*, 24(3):439–446, 1986.

[9] S. K. Gupta. n jobs and m machines job-shop problems with sequence-dependent set-up times. *International Journal of Production Research*, 20(5):643–656, 1982.

[10] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization.* John Wiley & Sons, New York, 1988.

[11] M. Queyranne and Y. Wang. Symmetric inequalities and their composition for asymmetric travelling salesman polytopes. *Mathematics of Operations Research*, 20(4):838–863, 1995.

[12] R. Z. Ríos-Mercado and J. F. Bard. New heuristics for the flow line problem with setup costs. *EJOR*, 1996. (Submitted).

[13] J. V. Simons Jr. Heuristics in flow shop scheduling with sequence dependent setup times. *OMEGA The International Journal of Management Science*, 20(2):215–225, 1992.

[14] B. N. Srikar and S. Ghosh. A MILP model for the n-job, m-stage flowshop with sequence dependent set-up times. *International Journal of Production Research*, 24(6):1459–1474, 1986.

[15] E. F. Stafford and F. T. Tseng. On the Srikar-Ghosh MILP model for the $N \times M$ SDST flowshop problem. *International Journal of Production Research*, 28(10):1817–1830, 1990.

[16] W. Szwarc and J. N. D. Gupta. A flow-shop with sequence-dependent additive setup times. *Naval Research Logistics Quarterly*, 34(5):619–627, 1987.