International Journal of
Operations Research

International Journal of Operations Research Vol. 3, No. 3, 155-172 (2006)

The $M \times N$ Flowshop Problem with Separable, Sequence-independent Setup Times

Edward F. Stafford, Jr.* and Fan T. Tseng

College of Administrative Science, University of Alabama in Huntsville, Room 355, Admin Sci Building, Huntsville, AL 35899, USA

Received April 2006; *Revised June 2006*; *Accepted July 2006*

Abstract—This paper presents an MILP model for the permutation flowshop wherein the setup times are both separable from the job processing times and independent of a job's position in the processing sequence. Two experiments were conducted to estimate the computer times necessary to solve problems with up to 9 machines and 15 jobs, and to then compare these solution time requirements to those required to solve the same sets of problems solved as regular (NSIST) flowshop problems. The resultant data were then used to assess the impact on two optimal sequence performance measures, makespan and mean flowtime when setup times were separated from their jobs and allowed to begin as soon as the machine was free from the preceding job. This impact of separated setup times was found to increase with increasing numbers of machines, but to decrease slightly with increasing numbers of jobs for a given number of machines. Lastly, the data were used to analyze the impact on mean flowtime when makespan is minimized, and the impact on makespan when mean flowtime is minimized.

Keywords—Integer programming, Flowshop, Setup times, Sequence-independent setup times, Makespan, Mean flow time

1. INTRODUCTION

The purposes of this paper are threefold: (1) to present and investigate a new MILP (mixed-integer linear programming) model for the $M \times N$ permutation flowshop with separable, sequence-independent setup times (SSIST); (2) to use this model to investigate the impact of SSIST on competing flowshop system performance measures (makespan and mean flow time); and (3) to use this model to investigate the tradeoffs between the competing performance measures as first one, then the other is optimized. First, to set the stage for this new model, the flowshop problem environment is described in some detail.

1.1 The classical flowshop problem

A flowshop consists of two major elements: (1) a set of N jobs; and (2) a series of M machines over which each job is to be processed in the same technological ordering of machines. As stated by Gupta (1972), this problem may be summarized as:

Given a set of N jobs to be processed on a series of M machines, find the ordering or sequencing of jobs that will minimize a well defined and desirable measure of production cost.

In this paper we further restrict the sequence to be permutation; that is, the jobs are processed on each machine in the exact same ordering. In his seminal paper on flowshop scheduling, Johnson (1954) presented many of the basic assumptions of this classical flowshop sequencing problem. A recent available source of all of these assumptions, divided into groups concerning the jobs (*J*1-*J*8), concerning the machines (*M*1-*M*5), and concerning operating policies (*P*1-*P*8), is Gupta and Stafford (2006).

Graves (1981) and others have shown that, except for a few special cases, such as Johnson's work with the two-machine regular flowshop and a few three-machine scenarios, all variants of the flowshop scheduling problem are computationally NP-complete or worse in complexity. Thus much of the research, to date, has concentrated on heuristic techniques for finding "good" or near-optimal solutions to many of these variants of the flowshop problem. At the same time, a variety of optimizing techniques has been successfully developed over the past forty years. These have included (with representational citations) complete or implicit enumeration (Aggarwal and Stafford, 1975), mathematical programming (Wagner, 1959; Stafford, 1988; Stafford et al., 2005), branch and bound (Ignall and Schrage, 1965; Lageweg, 1978) and combinatorial search (Baker, 1975; Gupta, 1975). This paper utilizes an MILP model, which combines the features of mathematical programming and branch and bound, to optimize the SSIST flowshop problem model described below.

1.2 Job setup time considerations

In their recent survey paper involving setup time considerations, Allahverdi et al. (1999) state: "The majority of scheduling research assumes setup as negligible or part of the processing time...." This assumption, labeled *J*6 by

*Corresponding author's email: staffoef@uah.edu

Gupta and Stafford, was one of the main elements of Johnson's description of the classical flowshop in 1954, and it carried through much of the flowshop sequencing literature up to the present. But for the past decade or so, more and more papers have been concerned with job setup time as an important element to consider. The impetus of this current paper is due, in part, to a recent spate of papers involving separable setup times by Aldowaisan and Allahverdi (1998), Allahverdi (2000), and Aldowaisan (2001).

Two aspects in considering job setup times, (1) the separability of setup times from the job processing times, and (2) the independence of a given job's setup time from the job immediately preceding that job in the processing sequence, are both major determinants of the scope of this paper.

1.2.1 Separability of setup times

Starting with Johnson (1954), most early flowshop sequencing researchers assumed that setup times were independent of a job's sequence position, and hence they could be included in the processing times. This inclusion effectively was an implicit assumption that setup times were ***not*** separable. Yoshida and Hitomi (1979) extended Johnson's work to two-machine flowshops wherein setup times were separable from processing times. As pointed out by these authors, realistic problems involve the situation wherein the setup for a job on a succeeding machine can be done before that job's completion on the preceding machine. Much of the subsequent research involving separable setup times has involved two- or three-machine flowshop problems.

The following example illustrates the impact of separable setup times on the scheduling performance measures makespan and mean flow time (see section 1.4, below, for a complete explanation of these measures). Assume there are two jobs, A and B, to be processed on a two-machine flowshop. The times (setup, processing) for job A on machines 1 and 2 are (1, 6) and (3, 5) respectively; for job B, (4, 6) and (3, 6). This A-B processing sequence is Gantt-charted in Figure 1 for both the non-separable and separable setup time cases. The makespan savings is 3 in the separable case (23 versus 26), and the mean completion time savings is 3 (17.5 versus 20.5). Analysis of this savings is an important aspect of the evaluation of the new MILP models presented later in this paper.

1.2.2 Independence of setup times

The great majority of flowshop scheduling research has included assumption *J*5 wherein the magnitude of each setup time is independent of the job's position in the sequence. That is, job *i*'s setup time on machine *r* is a constant regardless of the assigned position for job *i* in the processing sequence. When this setup time is not a constant, but rather varies depending on job *i*'s position in the job sequence, assumption *J*5 is modified, the problem becomes considerably more complex, and it is known as the SDST (sequence-dependent setup times) problem variant. Corwin and Esogbue (1974) were among the first to describe the SDST flowshop problem, which has proven amenable to MILP modeling as discussed below. Allahverdi et al. (1999) describe several real-life examples for the SDST flowshop variant.

1.2.3 A taxonomy of MILP flowshop models

Table 1 links these two aspects of setup time into a two-by-two taxonomy of MILP flowshop models. Each of the four cells of this table (NSI, SSI, NSD, SSD) contains two families of flowshop models, one each at the limits of the machine buffer sizes for the production system. Our acronyms for these models are of the form alpha/beta. The alpha portion consists of two parts: (1) an initial letter (*S* = separable, *N* = not separable) indicating the separability of the setup times from the job processing times; and (2) four additional letters (SIST = sequence-independent, SDST = sequence-dependent) indicating whether or not the values of these setup times are independent of the preceding job in the processing sequence. The beta portion of these acronyms is discussed below. (SSIST is equivalent to SIJST, and SSDST to SDJST in Chen et al. (1998).)

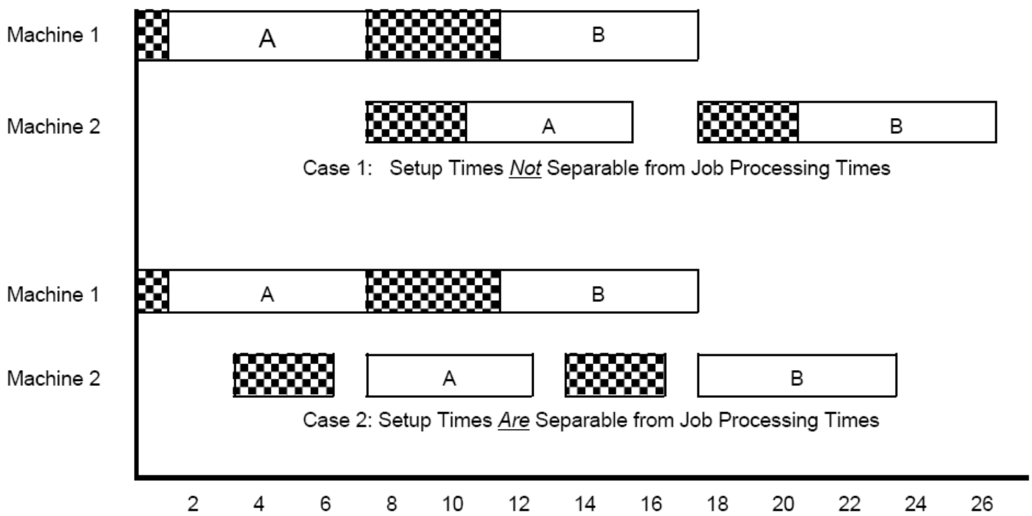

Figure 1. Separable versus non-separable setup times.

1.3 Interstage buffer considerations

Assumption *P*6 for the classical flowshop problem states:

Each machine is provided with adequate waiting space for allowing jobs to wait before starting their processing. (Gupta and Stafford, 2006)

This assumption of infinite-sized or *N*-sized queues or buffers at each machine in the processing system has been the predominant buffer assumption throughout the flowshop literature from 1954 to the present. At the same time, there has been an increasing number of papers examining flowshop problems with finite capacity queues and with no queues.

The extreme case is the "no wait" flowshop, also called the NIQ (no intermediate queues) flowshop (Wismer, 1972). In this variant, once a job begins its processing on machine 1, it must be processed without any delays on each of the *M* machines in the system. There are no buffers (except before machine 1) and a job may not sit idle on any machine, after it has finished processing on that machine, waiting for the job ahead of it in the sequence to finish processing on the next machine in the system. In their survey of "no wait" and "blocking" flowshop studies, Hall and Sriskandarajah (1996) identify a number of such flowshops occurring in industry because of process and/or product requirements.

The next case is to have no queues, but to allow a job to remain on the machine when it has completed its processing there if the predecessor job in the sequence is still being processed on the next downstream machine in the production system. This is called the "blocking" flowshop; the job of interest is blocked from advancing to the next machine by its sequence predecessor, and it may also be blocking its sequence successor from advancing upon completion of processing on an earlier machine. According to Table I in Hall and Sriskandarajah, studies of "blocking" flowshops have been limited to two-machine systems.

Table 1. Setup time taxonomy of MILP flowshop models

Setup Times	Sequence Independent	Sequence Dependent
Not Separable	NSIST/Reg[a] {Classical[b]}	NSDST/Reg
	(F_3/ /C_{MAX})[d] Wagner (1959)[c] (F_M/ /C_{MAX}, *MFT*) Stafford (1988) Stafford and Tseng (2002) Stafford et al (2005)	none
	NSIST/NIQ {NIQ}	NSDST/NIQ
	(F_M/*nwt*/C_{MAX}, *MFT*) Stafford (1988) Stafford and Tseng (2002)	none
Separable	SSIST/Reg	SSDST/Reg {SDST}
	(F_M/St_{si}/C_{MAX}, *MFT*) This Paper	(F_M/St_{sd}/C_{MAX}) Srikar and Ghost (1986) Rios-Mercado and Bard (1998) (F_M/St_{sd}/C_{MAX}, *MFT*) Tseng and Stafford (2001) Stafford and Tseng (1990,2002)
	SSIST/NIQ	SSDST/NIQ {SDST/NIQ}
	(F_M/*nwt*, St_{si}/C_{MAX}, *MFT*) Future Study	(F_M/*nwt*,St_{sd}/C_{MAX}, *MFT*) Stafford and Tseng (1990, 2002)

[a]Model acronym as described in this paper.
[b]Model synonym used in the literature.
[c]Authors and references.
[d]Lawler et al. (1993) three-field notation (shop type/shop conditions including setup information/performance criteria) as described by Allahverdi et al. (1999); F_M = flowshop with *M* machines; C_{MAX} = Makespan; *MFT* = mean flowtime; *nwt* = no wait; St_{si} = sequence independent setup time; St_{sd} = sequence dependent setup time

The third case involves flowshops with machine buffers of finite capacity. This problem was described as early as 1980 by Papadimitriou and Kanellakis (1980). Norman (1999) tested several heuristic procedures for solving this problem under a variety of assumptions about both the size of the buffers and the magnitude of the sequence-dependent setup times relative to job processing times. This case covers all models (except blocking) from the NIQ model (no buffers) to the regular flowshop wherein queue sizes are limited only by the number of jobs in the job set.

The second or "beta" portion of our notation (see Figure 1) may now be understood: Reg = regular flowshop; FB = finite buffer flowshop; ZB = "zero buffer" or "blocking" flowshop; and NIQ = flowshop with no intermediate queues, the "no wait" flowshop. This paper presents an MILP model for the first case: REG = regular flowshop. The NIQ, finite buffers and blocking models are left for development in future investigations. The development of the latter two models will result in an expansion of Table 1.

1.4 Sequence performance measures

Makespan is defined as the span of time required to "make" (process) all N jobs completely on the set of M machines. It is the time from when setup begins for the first job in the sequence on machine 1 to the time the last job in the sequence completes its processing on the last machine in the system. Makespan has been the predominant cost measure used in flowshop research, from Johnson (1954) to the present. It may be shown that minimizing makespan is equivalent to minimizing the sums of job processing times, machine idle times, and, where appropriate, separated setup times on machine M, the last machine in the system (Tseng and Stafford, 2001). For the NSIST/Reg (classical or regular) flowshop, Stafford (1988) showed that minimizing makespan is equivalent to minimizing machine idle time on the last machine.

Conway et al. (1967) and later researchers presented arguments for mean job flow time, also called mean flow time or mean job completion time, as a more appropriate measure of cost than makespan for evaluating flowshop scheduling techniques. Flowshop problems with mean flow time as the performance measure are NP-hard or worse for $M \geq 3$ machines (Graves, 1981). Job flow time, also called job completion time, is the difference in time between when a job first enters the production system until it completes processing on the last machine in the system. Based on assumption *J*1 (all jobs being available at time zero), flow time becomes the time when a job completes processing on machine M, C_{Mj}. Mean flow time then is the average of all job flow times, $\Sigma C_{Mj} / N$; but since N is a constant, minimizing ΣC_{Mj} is equivalent to minimizing mean flow time. This paper will use both makespan and mean job completion time as performance measures for all MILP models investigated.

1.5 Integer programming models for the flowshop problem

Table 1 lists the MILP literature relevant to the development of the MILP flowshop model described below. This new model will occupy the upper half of the currently empty lower left cell of this table. We conjecture that the flowshop problems represented by the currently empty upper right cell of Table 1 will be more appropriately modeled by a TSP (traveling salesman problem) technique than by MILP. Testing of this conjecture is left to a future study.

Wagner (1959) described an all-integer linear programming model for the classical regular flowshop (NSIST) with M = 3 machines. Stafford (1983, 1988) extended Wagner's model to the general M-machine case, converted it to an MILP model, added a set of constraints to insure that the first job in the sequence started on each machine in the production system at the earliest possible time, and developed five different performance measures, including makespan and mean flow time for this regular flowshop model. Stafford also modified his model to accommodate flowshops with the NIQ requirements (NSIST/NIQ). Kang and Markland (1988) used Stafford's NSIST/NIQ model to test various heuristic procedures for solving the NIQ flowshop problem. Stafford and Tseng (1990) proposed alternative models for the NSIST/Reg and NSIST/NIQ flowshop problems. These new models were derived from the SDST flowshop work of Srikar and Ghosh (1986). Stafford and Tseng (2002) found that the assignment problem approach (Stafford, 1988) was computationally superior to the dichotomous constraints approach of Manne (1960) for both the NSIST Regular and NIQ flowshop problems. Further, this computational efficiency increased with increasing problem size in both N and M.

Srikar and Ghosh (1986) presented a dichotomous constraints MILP model for the SDST (SDST/Reg) flowshop problem. Stafford and Tseng (1990) made some minor corrections to the Srikar-Ghosh model and demonstrated that the modified model was robust to violations of the triangular inequality assumptions of dependent setup times. They also proposed alternative models for the NSIST/Reg, NSIST/NIQ and SDST/NIQ flowshop problems based on their investigation of the Srikar-Ghosh model. Rios-Mercado and Bard (1998) proposed an alternative MILP model for the SDST/Reg flowshop problem, and then showed that the Stafford-Tseng modification of the original Srikar-Ghosh model dominated their model for both branch-and-bound and branch-and-cut versions of MILP solution techniques. Tseng and Stafford (2001) described two new models for the SDST/Reg flowshop problem, and Stafford and Tseng (2002) added a new model for the SDST/NIQ problem. Their work included comparisons of these new, assignment-problem based models with their earlier dichotomous constraint models. The new SDST/Reg models were shown to be computationally superior to the earlier models, for all problem sizes tested. For the

SDST/NIQ problem, the new models were computationally superior for problems with $N \geq 8$ jobs.

In the forgoing paragraphs, we have set the stage for the presentation and analysis of a new MILP model for flowshops with separable, sequence-independent setup times (SSIST). This new model is presented in section 2 of this paper. In section 3, we use this new model to address a number of research questions related to the impact of allowing separable setup times, and related to the relationships between makespan and mean job flow time as system performance measures. In section 4, we provide a summary, conclusions drawn from this work, and a list of projects meriting future research efforts.

2. MILP MODEL FOR THE SSIST FLOWSHOP PROBLEM

In this section, we first present the new mixed-integer linear programming (MILP) model for the permutation flowshop wherein the job setup times are both separable from job processing times and independent of the job's position in the processing sequence. We refer to this model as the SSIST flowshop model. We then present the WST/Reg model reported by Stafford and Tseng (2002) for the regular flowshop wherein setup times are assumed to be included with the job processing times. We refer to this model here as the NSIST flowshop model.

2.1 Model notation

The notational conventions used in this paper follow that of Stafford and Tseng (1990, 2002). The subscript symbols used are: i for jobs, ($1 \leq i \leq N$); j for sequence position, ($1 \leq j \leq N$); and r for machines, ($1 \leq r \leq M$). The variables are defined as follows:

- X_{rj} idle time on machine r before the start of job in sequence position j
- Y_{rj} idle time of job in sequence position j after it finishes processing on machine r
- Z_{ij} 1, if job i is assigned to sequence position j, 0 otherwise
- C_{Mj} completion time of job is sequence position j on the last machine of the M-machine production line

The C_{Mj} variables are used to formulate the objective functions. The Z_{ij} are binary integer variables. The others are real variables that take integer values when processing times are also given integer values. Hence both models described below are MILP models.

There are two sets of parameters for the SSIST model, one each for job processing times and job setup times. The $M \times N$ matrix $\boldsymbol{P} = \{P_{ri}\}$ represents the known or computable processing times of all jobs on all M machines, where P_{ri} is the processing time for job i on machine r. Further, the $M \times N$ matrix $\boldsymbol{S} = \{S_{ri}\}$ represents the setup times of all jobs on these same M machines, where S_{ri} is the separable, sequence-independent setup time of job i on machine r. In the classical flowshop problem, called here the NSIST problem, it is assumed that setup times are sequence-independent, and that they can be added to the job processing times. Thus, for the NSIST model, $\boldsymbol{T} = \boldsymbol{P} + \boldsymbol{S} = \{T_{ri}\}$, where T_{ri} is the total time (setup plus processing) required for job i on machiner.

2.2 The SSIST model

The MILP model for the SSIST regular flowshop was developed using the assignment problem approaches of Wagner (1959) and Stafford (1988).

2.2.1 Constraint equation

The equation for the four sets of constraints of the SSIST model are as follows:

$$\sum_{j=1}^{N} Z_{ij} = 1 \qquad \{i = 1, ..., N\} \tag{1}$$

$$\sum_{i=1}^{N} Z_{ij} = 1 \qquad \{j = 1, ..., N\} \tag{2}$$

$$\sum_{i=1}^{N} (S_{ri} + P_{ri} - S_{r+1,\,i}) Z_{i,\,j+1} - \sum_{i=1}^{N} P_{r+1,\,j} Z_{ij} + (X_{r,\,j+1} - X_{r+1,\,j+1}) + (Y_{r,\,j+1} - Y_{rj}) = 0 \quad \{r = 1, ..., M-1; j = 1, ..., N-1\} \tag{3}$$

$$\sum_{i=1}^{N} (S_{ri} + P_{ri} - S_{r+1,\,i}) Z_{i1} + (X_{r1} - X_{r+1,\,1}) + Y_{r1} = 0 \quad \{r = 1, ..., M-1\} \tag{4}$$

Eq. (1) and (2), which control the assignment of jobs to positions in the job processing sequence, represent $2N$ constraints that are identical in form and purpose to the constraints of the classical assignment problem. Only one of the decision variables in each of these $2N$ equation is nonzero at one time because all of the Z_{ij} variables are binary integers. Eq. (1) insures that each job i is assigned to one, and only one position j in the sequence, while Eq. (2) insures that one, and only one job is assigned to each position in the sequence.

The $(M-1) \times (N-1)$ constraints represented by Eq. (3) insure the following: (1) the job in position j in the sequence cannot begin to be processed on machine $r+1$ until it has completed its processing on machine r, and its setup on machine $r+1$ has been completed; and (2) the setup for the job in position $j+1$ in the sequence cannot be performed on machine r until the processing of the job in position j in the sequence has been completed on this same machine. Stafford and Tseng (1990, 2002) referred to this type of constraint set as JAML (job-adjacency-machine-linkage) constraints.

The $(M-1)$ constraints represented by Eq. (4) control the start time of the first job in the sequence on machines 2 through M. The earliest starting time for this first job on machine 1 is immediately after the first machine has been set up. The starting time of this first job on subsequent machines, 2 through M, is determined by the maximum of (1) the completion time of this job on the previous machine, $r - 1$; and (2) the setup time on the current machine r. If the completion time on the previous machine occurs first, then this first job has idle time $Y_{r-1,1}$ before it can proceed on the current machine r. Otherwise, machine r has idle time X_{r1} while awaiting the first job to be completed on the previous machine. The derivation of Eq. (3) and (4) is provided in the Appendix of this paper.

In addition to the four sets of constraints described above, the following set of relationships is used to represent the completion time of the jobs in each of the N sequence positions on Machine M, the last machine in the production system.

$$C_{Mj} = \sum_{p=1}^{j}\sum_{i=1}^{N}(P_{Mi} + S_{Mi})Z_{ip} + \sum_{p=1}^{j} X_{Mp}$$

$$\{j = 1, ..., N\} \qquad (5)$$

Eq. (5) does not actually constrain the integer programming problem; rather, it is used as shown below to develop the objective functions for this model.

2.2.2 Objective functions

From Johnson (1954) onward, minimizing makespan has been the predominant performance measure optimized for the flowshop problem. Since all jobs are assumed to be available at time = 0, jobset makespan is the completion time of the last job in the sequence on the last machine in the production system. That is:

$$\text{Makespan} = F_{MAX} = C_{MN} \qquad (6)$$

Again, since all jobs are available at time = 0, mean flowtime is the average completion time of all N jobs on the last machine. This may be expressed as:

$$\text{Mean flowtime} = F_{BAR} = \sum_{j=1}^{N} C_{Mj} / N \qquad (7)$$

Recently, a number of authors (Aldowaisan and Allahverdi, 1998) have reported flowshop studies that used total flowtime as the objective. Eq. (7) may be easily modified to represent total flowtime as follows:

$$\text{Total flow time} = F_{TOT} = \sum_{j=1}^{N} C_{Mj} \qquad (8)$$

Stafford (1988) proposed three other possible objectives to minimize for the flowshop: (1) total job idle time, while the jobs wait for the next machine in the processing sequence to be ready to process them; (2) total overall machine idle time; and (3) within-sequence machine idle time. These are represented, respectively, by the following equations:

$$\text{Total job idle time} = \sum_{i=1}^{N}\sum_{r=1}^{M} Y_{ri} \qquad (9)$$

$$\text{Overall total machine Idle Time} = \sum_{i=1}^{N}\sum_{r=1}^{M} X_{ri} \qquad (10)$$

$$\text{Within-Sequence Machine Idle Time} = \sum_{i=2}^{N}\sum_{r=1}^{M} X_{ri} \qquad (11)$$

The SSIST MILP model may be summarized as:

Minimize: (6), (7), (8), (9), (10) or (11), subject to (1), (2), (3), (4), and (5).

2.3 The NSIST model

The WST/Reg model of Stafford and Tseng (2002) was chosen to represent the regular permutation flowshop for investigating the impact of the SSIST assumptions on various aspects of the flowshop problem. This model was chosen because it was the impetus for developing the new SSIST model, and because it is quite similar to this SSIST model. The variables and parameters for this model are as defined above. For this paper, we call this model the NSIST flowshop model.

The NSIST model also uses the classical assignment problem constraints, Eq. (1) and (2) above, to control the assignment of jobs to positions in the job processing sequence. From above, $T_{ri} = P_{ri} + S_{ri}$. The remaining sets of constraints for the NSIST model are then as follows:

$$\sum_{i=1}^{N} T_{ri} Z_{i,j+1} - \sum_{i=1}^{N} T_{r+1,i}\ Z_{ij} + X_{r,j+1} - X_{r+1,j+1} + Y_{r,j+1}$$
$$-Y_{rj} = 0 \quad \{r = 1, ..., M-1; j = 1, ..., N-1\} \qquad (12)$$

$$\sum_{i=1}^{N} T_{ri} Z_{i1} + X_{rl} - X_{r+1,1} + Y_{r1} = 0 \quad \{r = 1, ..., M-1\} \qquad (13)$$

$$Y_{r1} = 0 \quad \{r = 1, ..., M-1\} \qquad (14)$$

$$C_{Mj} = \sum_{p=1}^{j}\sum_{i=1}^{N} T_{Mi} Z_{ip} + \sum_{p=1}^{j} X_{Mp} \quad \{j = 1, ..., N\} \qquad (15)$$

Eq. (12) and (13) serve the same function as Eq. (3) and (4) respectively in the SSIST model. Eq. (14) is required to anchor the first job in the processing sequence to the zero time line of the Gantt chart (see Stafford, 1988). Eq. (15) is the NSIST equivalent to Eq. (5) of the SSIST model. The NSIST model may then be summarized as:

Minimize: (6), (7), (8), (9), (10), or (11), subject to (1), (2), (12), (13), (14) and (15).

The SSIST and NSIST models are extensively compared and investigated in the next section of this paper.

3. MODEL ANALYSES

In this section, we address a number of research questions regarding the SSIST flowshop problem model described above. One question is posed with regard to model size complexity:

Q1 What is the size complexity of the MILP model for the SSIST flowshop, and how does this complexity compare with the size complexity of the comparable model for the NSIST flowshop?

The following research questions are posed with regard to computer solution time requirements:

Q2 What are the time requirements for solving problems of various sizes for the SSIST flowshop model?

Q3 How do these solution time requirements compare to the requirements for the corresponding model in the NSIST environment, and to similar studies in the literature?

With regard to the impact of separating setup times from job processing times, the following research question is posed:

Q4 What is the impact on sequence performance for the regular flowshop problem when the job setup times are separated from job performance times?

For Q4, this question is investigated for both job sequence makespan and for mean job flowtime.

Q5 What is the impact of minimizing one performance measure (makespan or mean flowtime) on the other performance measure?

Question Q5 addresses the tradeoffs between the two major performance measures prevalent in the flowshop scheduling literature. We first address the issue of size complexity with regard to research question Q1. The remainder of this section is dedicated to investigating research questions Q2 through Q5.

3.1 Size complexity of the models

The size complexity of the SSIST flowshop model and its corresponding NSIST model, as a function of M (number of machines) and N (number of jobs), is shown in Table 2. This complexity is divided into two parts: (1) number of decision variables; and (2) number of constraints. Both models require N^2 binary integer variables to assign the N jobs to the N positions in the processing sequence. Both models require an additional $2MN + 2$ real variables to account for machine and job idleness, job completion times, makespan, and mean flow time. The SSIST model requires $2N + MN + 2$ constraints, while the NSIST mode requires the same number plus an additional $M - 1$ constraints due to Eq. (13).

3.2 Computer experiments

Two full factorial experiments were designed and executed to help answer research questions Q2 through Q5 posed above. The first experiment, conducted several years ago, utilized LINDO (LINDO Systems, 1999) to conduct an initial investigation of these questions. The second experiment, conducted recently, utilized LINGO (LINDO Systems, 2004) and a newer computer to extend the size of problems solvable with both the SSIST and NSIST models.

Table 2. Size complexity of SSIST and NSIST flowshop models

Model	SSIST	NSIST
Variables		
Integer $\{Z\}$	N^2	N^2
Real $\{X, Y, C\}$	$2MN$	$2MN$
Other ‡ $\{F_{MAX}†, F_{BAR}†\}$	2	2
Total Variables	$N^2 + 2MN + 2$	$N^2 + 2MN + 2$
Constraints		
Assignment {(1), (2)}	$2N$	$2N$
JAML {(3), (4); (11), (12)}	$N(M - 1)$	$N(M - 1)$
Job Idle {(13)}	- - - -	$M - 1$
Completion Time {(5); (14)}	N	N
Other ‡ {(6), (7)}	2	2
Total Constraints	$2N+MN+2$	$2N+MN+M+1$

†F_{MAX} = makespan; F_{BAR} = mean job completion or flow time.
‡Two variables and constraints added to simplify measuring performance measures.

3.2.1 Experiment I, using LINDO

The two major factors in this experiment were M (number of machines) and N (number of jobs). Factor levels were M = 5, 7, and 9; and N = 6, 7, 8, and 9. These levels correspond to the earlier experiments reported by Tseng and Stafford (2001), Stafford and Tseng (2002), and Tseng et al. (2004). There were five replications for each of the 12 cells of the experiment. The job processing times and the job setup times for each problem generated were both uniformly distributed: $P_{ri} \in U[1, 100]$ and $S_{ri} \in U[1, 25]$. This distribution of setup times is in line with the 20-40% reported by Gupta and Darrow (1986) for S_{ri}/P_{ri} ratios found in practice. Each of the sixty base problems was then processed through four different problem generators to create four different flowshop problems: [SSIST or NSIST] × [makespan or mean flow time].

The resultant 240 flowshop problems were each solved, one at a time, with version 6.1 of Hyper LINDO on a Dell Pentium III 800 MHz microcomputer equipped with 128 MB of RDRAM. The specific LINDO options utilized were integer programming, an optimality stopping criterion of zero, terse output, and the "Take" command. The LINDO elapsed time feature was used to measure actual problem solution times to the nearest 0.01 second. The optimizing criteria used were minimizing job set makespan for 120 problems and minimizing mean job completion time for the other 120 problems.

3.2.2 Experiment II, using LINGO

This experiment used the same major factors, M and N, that were used in Experiment I. Factor levels were M = 5, 7, and 9; and N = 10, 11, 12, 13, 14, and 15. The job processing times and the job setup times were again both uniformly distributed: $P_{ri} \in$ U[1,100] and $S_{ri} \in$ U[1,25]. There were five replications for each of the 18 cells of this experiment; and each of the resultant 90 problems was represented by a simple data set that could be read individually, or in batches, by all of the four LINGO programs that were written, one for each of the four major cells of this experiment: [SSIST or NSIST] × [makespan or mean flow time].

All 360 problem instances were solved using LINGO 9.0 on a Dell Pentium IV 2 GHz personal computer equipped with 512 MB of RDRAM. Specific LINGO options employed were integer programming, terse output, an optimality stopping criterion of zero, and script files to solve the problems in batches of 30. The LINGO elapsed time option was used to measure actual problem solution times to the nearest 0.01 second. The optimizing criteria again were minimizing job set makespan for 170 problems and minimizing mean job completion time for the other 170 problems. (The $M \times N = 9 \times 15$ cell problems for experiment II were not run due to excessive computer solution time requirements for these problems.)

For both experiments I and II, the relevant solution values were summarized, processed, and compared in a large Excel workbook. The results of these comparisons are presented below.

3.3 Computer solution times

The computer solution times (in seconds) for each of the 12 cells of experiment I and 17 of the 18 cells of experiment II are summarized in Tables 3 and 4 respectively. The mean solution times are given for each cell of five problems, for both the makespan and mean flowtime performance measures, and for both the SSIST and NSIST flowshop models. Following Stafford et al. (2005), the cell median solution time values are also reported for both experiments. The computer solution times for the two experiments cannot be directly linked and compared because each experiment was executed with different MILP software.

Table 3. Computer solution time means and medians, experiment I

Minimize Makespan						Criterion		Minimize Mean Flow Time					
Mean Solution Time			Median Solution Time					Mean Solution Time			Median Solution Time		
SSIST	NSIST	SSI/NSI‡	SSIST	NSIST	SSI/NSI‡	*N*‡	*M*‡	SSIST	NSIST	SSI/NSI	SSIST	NSIST	SSI/NSI
0.26†	0.28	0.923	0.22†	0.27	0.815	6	5	0.14	0.25	0.567	0.11	0.22	0.500
0.75	0.93	0.800	0.66	0.88	0.750		7	0.65	0.57	1.137	0.61	0.50	1.220
1.12	0.91	1.228	1.26	0.83	1.518		9	0.95	1.06	0.889	1.04	1.10	0.945
0.71	0.71	1.000	0.62	0.71	0.873	Mean/Med.§		0.58	0.63	0.921	0.55	0.49	1.122
0.48	0.26	1.860	0.42	0.17	2.471	7	5	0.59	0.58	0.876	0.60	0.44	1.364
1.78	1.44	1.233	1.75	3.68	0.476		7	1.91	2.33	0.820	2.22	2.42	0.909
2.70	3.80	0.729	1.92	3.51	0.547		9	3.25	3.13	1.039	3.13	3.24	0.966
1.68	1.83	0.914	1.54	1.43	1.077	Mean/Med.		1.92	2.01	0.938	1.65	2.14	0.771
0.89	1.40	0.636	1.21	1.09	1.110	8	5	1.56	1.83	0.852	1.32	1.48	0.892
6.93	5.96	1.162	5.06	3.68	1.375		7	7.29	5.10	1.430	7.97	4.45	1.791
13.46	12.49	1.078	10.33	14.39	0.718		9	10.33	11.06	0.934	9.39	8.51	1.103
7.09	6.62	1.072	4.55	3.63	1.253	Mean/Med.		6.40	6.00	1.066	4.99	4.45	1.121
4.51	6.31	0.715	3.13	4.95	0.632	9	5	5.79	5.51	1.050	5.88	7.41	0.794
15.88	15.74	1.009	12.53	16.26	0.771		7	14.79	12.60	1.175	13.29	10.38	1.280
52.14	53.44	0.976	47.84	35.37	1.353		9	55.47	55.45	1.000	57.24	66.18	0.865
24.17	25.16	0.961	12.53	16.26	0.771	Mean/Med.		25.35	24.52	1.034	13.29	10.38	1.280
8.41	8.58	0.980	1.81	1.98	0.914	Overall§		8.56	8.29	1.033	2.74	2.59	1.060

†Mean or median computer solution time for 5 problems, seconds.
‡N=number of jobs; M=number of machines; SSI/NSI= ratio of SSIST mean solution time to NSIST mean solution time.
§ "Mean/Med."= average or median of all problems for a given N value; "Overall"= average or median for all problems in experiment.

Table 4. Computer solution time means and medians, experiment II

Minimize Makespan						Criterion		Minimize Mean Flow Time					
Mean Solution Time			Median Solution Time					Mean Solution Time			Median Solution Time		
SSIST	NSIST	SSI/NSI‡	SSIST	NSIST	SSI/NSI‡	*N*‡	*M*‡	SSIST	NSIST	SSI/NSI	SSIST	NSIST	SSI/NSI
13.09†	10.09	1.298	9.41	9.96	0.945	10	5	10.67	11.09	0.962	9.70	11.99	0.809
20.44	18.79	1.088	15.92	13.00	1.225		7	22.63	20.50	1.104	23.52	21.19	1.110
56.95	40.04	1.422	29.85	33.67	0.887		9	26.06	38.23	0.682	27.37	37.92	0.722
30.16	22.97	1.313	15.92	15.33	1.038	Mean/Med.§		19.79	23.27	0.850	20.85	21.19	0.984
12.68	13.05	0.972	11.85	9.91	1.196	11	5	16.74	27.18	0.616	18.04	33.32	0.541
40.81	19.55	2.087	42.60	19.97	2.133		7	38.36	47.53	0.807	34.26	48.61	0.705
223.18	145.47	1.534	100.55	72.95	1.378		9	118.40	95.61	1.238	70.67	77.79	0.908
92.22	59.36	1.554	42.40	23.51	1.803			57.84	56.78	1.019	34.26	48.61	0.705
16.84	12.36	1.352	14.36	7.88	1.822	12	5	35.93	47.48	0.757	34.09	46.03	0.741
44.86	22.48	1.340	46.81	18.82	2.487		7	92.95	103.37	0.899	78.82	71.95	1.095
253.36	316.62	0.803	114.47	308.94	0.371		9	666.06	827.68	0.805	668.61	656.03	1/019
105.02	120.49	0.872	46.81	34.35	1.363	Mean/Med.		264.98	326.18	0.812	78.82	80.57	0.978
99.32	127.21	0.781	28.36	75.67	0.375	13	5	406.09	212.44	1.912	262.99	256.77	1.024
86.93	131.52	0.661	80.57	103.05	0.782		7	601.80	811.28	0.742	521.55	452.04	1.154
721.05	466.80	1.545	187.84	410.21	0.458		9	616.77	708.02	0.871	634.73	687.68	0.923
302.43	241.84	1.251	95.52	139.71	0.684	Mean/Med.		541.55	577.25	0.938	521.55	344.92	1.512
36.24	93.11	0.389	27.19	53.36	0.510	14	5	461.94	434.81	1.062	387.47	448.61	0.864
1091.777	7232.312	0.151	610.83	808.18	0.756		7	1869.64	1686.23	1.109	1408.69	856.30	1.645
6688.50	5777.82	1.158	3403.28	3110.37	1.094		9	7905.346	9643.32	0.820	4800.19	3438.91	1.396
2605.50	4367.75	0.597	285.71	534.63	0.534	Mean/Med.		3412.32	3921.45	0.870	1408.69	989.32	1.424
13.50	17.312	0.780	2.96	3.73	0.794	15	5	861.208	2014.16	0.428	782.17	476.06	1.643
6308.10	6072.794	1.039	395.19	1026.80	0.385		7	10035.912	9617.37	1.044	4220.71	4788.92	0.881
3160.80	3045.0533	1.038	170.07	109.88	1.548	Mean/Med.		5448.560	5815.77	0.937	1615.32	2603.48	0.620
925.15	1207.49	0.766	46.81	42.03	1.114	Overall§		1399.208	1549.78	0.903	99.45	138.51	0.718

†‡§See Table 3 for footnote explanations.

3.3.1 Solution times for the SSIST model (Q2)

For experiment I, both cell solution time summary measures -- mean and median -- are consistent. That is, for increasing M (within cells) and for increasing N (between adjacent cells), both the means and medians increase without fail for both performance measures -- makespan and mean flowtime. With a single exception, this consistency holds in Experiment II for the mean flowtime criterion. For both cell mean and median solution times, the values for cell $M \times N = 9 \times 12$ are larger than the corresponding values for cell 9×13. This single inconsistency may be explained in large part by examining the individual problem solution times in the two cells. Problem 091202 required 1045 seconds to solve to optimality, a value approximately 83% larger than the mean for the other four problems in that cell. Problem 091300, on the other hand, required just 353 seconds to solve, approximately 57% less time than the mean for the other four problems in that cell. (The problem number notation is of the form *mm-jj-pp* where *mm* = number of machines in the flowshop, *jj* = number of jobs in the problem, and *pp* = problem number, 00, 01, ...) These occurrences of a relatively quick solution time (problem 091300) and a relatively slow solution time (problem 091202) are not unique to this study; similar occurrences have been reported by Tseng and Stafford (2001), Stafford and Tseng (2002), and Stafford et al. (2005), among others.

For the makespan criterion in experiment II, the results are not nearly so consistent as those described above. For mean solution times, the mean of $M \times N = 5 \times 10$ is greater than that of cell 5×11. This difference is explained by the result that problem 051103 required just 1.16 seconds to solve to optimality compared to the mean time of 15.58 seconds for the other problems in that cell. The 5×13, 5×14, and 5×15 cells exhibit a major inconsistency with cell means of 99.32, 36.24, and 13.50 seconds respectively. Cell 5×13 has two problems (0 and 4) with a mean solution time more than 20 times that of the other three problems in that cell. The solution times for cell 5×15 are quite unusually small with three solution times under 3 seconds. We attribute this solution time inconsistency to these two large solution times (cell 5 × 13) and three very fast solution times (cell 5 × 15). Further, this explanation holds for the inconsistency among these same three cells for the median cell solution times. The inconsistency between cells 5 × 13 and 7 × 13, for makespan, is also explained by the very short solution tome for problem 051103. The slight inconsistency between cells 5 × 13 and 5 × 14 for the median solution times may also be explained by the two problems in cell 5 × 13 that require extensive time to solve.

The overall mean and median values in Tables 3 and 4 suggest that, on average, the SSIST model solves a problem in less time when makespan is the performance measure than when mean flowtime is the performance measure. For experiment I, the mean solution times for makespan and mean flowtime were 8.41 seconds and 8.56 seconds, respectively; the median solution times were 1.81 and 2.74 seconds, respectively. The makespan measure required less time than the mean flowtime measure for 34 of the 60 problems, the mean flowtime measure required less time for 25 problems, and there was one tie. For experiment II, the respective overall means were 925.15 seconds and 1399.20 seconds for minimizing makespan versus mean flowtime; and the respective median solution times were 46.81 seconds and 99.45 seconds. The makespan solutions were faster in 61 of the 85 problems for experiment II, while the mean flowtime solutions were faster for 24 of the problems. It should be noted that Eq. (7), mean flowtime, was used in experiment I while Eq. (8), total flowtime, was used in experiment II as a surrogate for mean flowtime. It is possible that this alternative although equivalent objective function in experiment II impacted on the improved success of makespan versus mean flowtime in this experiment.

3.3.2 Solution times: SSIST vs NSIST models (Q3)

The means and medians summary values in Tables 3 and 4 suggest that, overall, there is no clear difference between the SSIST and NSIST models regarding required computer solution times for two common sets of problems. For experiment I, the overall mean and median solution times were slightly smaller for the SSIST model when makespan was the optimized performance measure; but these same measures were slightly higher for the SSIST model when mean flowtime was the performance measure. The data in Table 5 indicate that the SSIST model was faster than the NSIST model in 65 of the 120 total problem instances in experiment I.

In experiment II, for the makespan performance measure, the overall mean solution time for the SSIST model was significantly smaller than that for the NSIST model (925.15 vs 1207.49 seconds), yet the median solution time for the SSIST model was higher than that for the NSIST model (46.81 vs 42.03 seconds). When mean flowtime is the optimizing performance measure, both the overall mean and median solution times were less for the SSIST model than for the NSIST model. From Table 5, the SSIST model was faster in 88 of the 170 total problem instances for experiment II. The "*Sseq*" column data in Table 5 indicate the number of problems in which the SSIST and NSIST models found exactly the same optimal sequence for each performance measure in each experiment.

Table 5. Miscellaneous data, experiments I and II

Measure	*SGN*†	*NGS*	*SSeq*†	Total # Problems
Experiment I				
F_{MAX}‡	28	32	13	60
F_{BAR}‡	27	33	22	60
Total	55	65	35	120
Experiment II				
F_{MAX}‡	43	42	6	85
F_{BAR}‡	39	46	9	85
Total	82	88	15	170
Grand Total	137	153	50	290

‡F_{MAX} = makespan; F_{BAR} = mean flow time.
†*SGN* = # problems wherein **S**SIST model solution time **g**reater than **N**SIST model solution time; *NGS* = reverse;
Sseq = # problems both models having same optimal sequence.

3.3.3 Solution times SSIST and NSIST vs the literature (Q3)

The SSIST model reported in this paper is, to our knowledge, the first model for the *M* × *N* flowshop with separable sequence-independent setup times reported in the literature; and as such, it seems reasonable to benchmark this model to similar models in the literature. Stafford et al. (2005) included the WST model in their comprehensive analysis of two families of MILP models for the regular flowshop; and this WST model is quite close to the SSIST model in structure and size. These authors also ran two experiments, each of which is equivalent to experiments I and II of the current paper; and the Pentium IV computer they used is the same machine used in experiment II of the current paper. Table 6 presents the summary data for the SSIST model, for both performance measures and both experiments, as well as the summary data from the two Stafford et al. experiments. Overall, the SSIST times compare well with the WST model results. It is expected that the WST model computer solution times would differ from those times for the SSIST solution times since Stafford et al. ran their LINDO solutions on a Pentium IV computer and a Pentium III computer was used for experiment I in this paper

3.4 Impact of separable setup times on performance measures

We next address the second purpose of this paper, that of using this model to investigate the impact of SSIST on competing flowshop system performance measures. This impact is described in the following two sub-sections, for makespan and mean flow time respectively.

3.4.1 Impact on makespan (Q4)

The left half of Table 7 presents summary data for investigating the impact of separable setup times on makespan for the problem sizes solved in this paper. For each $M \times N$ cell in the combined experiments, the mean makespan for the SSIST model, the mean makespan for the regular flowshop model (NSIST), the differences between these means, and the ratio of the mean NSIST makespan to the mean SSIST makespan are given.

The NSI/SSI ratio in Table 7 indicates the average penalty when setup times cannot begin until a job arrives at a machine (NSIST) relative to allowing setup times to commence as soon as the machine is clear of the previous job in the sequence. For the experimental cells based on number of machines M, the overall penalty for M = 5, 7, and 9 machines increases, with values of 5.4, 8.0, and 9.0 percent, respectively; the overall impact is an additional 7.4 percent makespan time required.

Within each M-value cell there is a clear pattern for the NSI/SSI values. As the number of jobs N increases from 5 to 15, this ratio steadily decreases for each value of M. The intercept and slope values for simple linear regressions of the NSI/SSI ratios as a function of N (number of jobs) are shown for each value of M, for the makespan performance measure, in the top half of Table 8. The Pearson r-values indicate a very good fit for the data for each M-value.

Table 6. Comparison of SSIST and WST§ models solution times

Experiment I					Experiment II				
M‡	N‡	SSIST/F_{MAX}‡	SSIST/F_{BAR}‡	WST/F_{MAX}‡§	M‡	N‡	SSIST/F_{MAX}‡	SSIST/F_{BAR}‡	WST/F_{MAX}‡§
5	6	0.26†	0.14	0.24	5	10	13.09	10.67	31.25
	7	0.48	0.59	0.63		11	12.68	16.74	20.66
	8	0.89	1.56	1.54		12	16.84	35.93	138.56
	9	4.51	5.79	3.02		13	99.32	406.09	18.87
						14	36.24	461.94	57.69
						15	13.50	861.21	1169.25
		1.54	2.02	1.36	Average		31.95	298.76	239.25
7	6	0.75	0.65	0.58	7	10	20.44	22.63	19.20
	7	1.78	1.91	1.59		11	40.81	38.36	2541.75
	8	6.93	7.29	3.61		12	44.86	92.95	140.75
	9	15.88	14.79	23.88		13	86.93	601.80	351.95
						14	1091.77	1869.68	820.07
						15	6308.10ƒ	10,035.91	**853.46**
		6.33	6.16	7.42	Average		256.96ƒ	525.08ƒ	790.77ƒ
9	6	1.12	0.95	1.34	9	10	56.95	26.06	26.48
	7	2.77	3.25	3.40		11	223.18	118.40	379.98
	8	13.46	10.33	11.53		12	253.36	666.06	248.19
	9	52.14	55.47	27.35		13	721.05	616.77	1674.15
						14	6680.50	7905.34	8994.67
						15	- - -	- - -	18,477.10
		17.37	17.50	10.92	Average		1588.61ƒ	1866.53ƒ	2264.69ƒ
		8.41	8.56	6.57	Overall		624.17ƒ	914.97ƒ	1107.14ƒ

‡F_{MAX} = makespan criterion; F_{BAR} = mean flow time criterion; M = # of machines; N = # of jobs.
†Mean computer solution time of five problems, seconds.
§Model WST and data from Stafford et al. (2005).
ƒMeans do not include values from N = 15 cell(s); in boldface cell, one problem did not solve in >13h.

Table 7. Impact of separable setup times on makespan and mean flow time

Minimize Makespan				Objective		Minimize Mean Flowtime			
SSIST	NSIST	NSI – SSI§	NSI/SSI§	*M*§	*N*§	SSIST	NSIST	NSI – SSI§	NSI/SSI§
612.4†	668.4†	56.0†	1.091†	5	6	410.20‡	459.77‡	49.57‡	1.121‡
648.4	685.4	37.0	1.057		7	426.80	473.38	46.58	1.109
782.8	830.2	47.4	1.061		8	494.75	550.85	56.10	1.113
824.8	872.4	47.6	1.058		9	524.18	575.62	51.44	1.098
894.4	936.6	42.2	1.047		10	557.10	606.28	49.18	1.088
977.2	1027.4	50.2	1.051		11	582.31	634.22	51.91	1.089
1041.8	1093.8	52.0	1.050		12	625.63	675.82	50.18	1.080
1037.0	1083.2	46.2	1.045		13	621.82	668.95	47.14	1.076
1048.2	1091.6	43.4	1.041		14	617.33	666.84	49.51	1.080
1252.4	1304.2	51.8	1.041		15	735.92	792.03	56.11	1.076
911.94	959.32	47.38	1.054	Average		559.60	610.38	50.77	1.093
719.0	808.4	89.40	1.124	7	6	514.77	593.37	78.60	1.153
788.2	874.2	86.00	1.109		7	583.80	664.37	80.57	1.138
871.2	948.0	76.80	1.088		8	592.50	666.95	74.45	1.126
954.8	1027.6	72.80	1.076		9	652.49	725.27	72.78	1.112
1008.8	1091.0	82.2	1.081		10	665.72	740.24	74.52	1.112
1030.0	1100.6	70.6	1.069		11	667.82	745.93	78.11	1.117
1125.6	1201.0	75.4	1.067		12	700.27	792.07	91.80	1.131
1169.8	1254.8	85.0	1.073		13	766.68	844.28	77.60	1.101
1260.4	1335.4	75.0	1.060		14	791.97	865.91	73.94	1.093
1312.0	1386.0	74.0	1.056		15	818.41	892.53	74.12	1.091
1023.98	1102.70	78.72	1.080	Average		675.44	753.09	77.65	1.117
908.8	1007.2	98.40	1.108	9	6	678.87	775.70	96.83	1.143
912.6	1020.0	107.40	1.118		7	677.66	779.71	102.06	1.151
1014.0	1114.0	100.00	1.099		8	721.05	824.95	103.90	1.144
1065.8	1160.8	95.00	1.089		9	776.40	872.42	96.02	1.124
1145.2	1240.0	94.8	1.083		10	788.74	897.16	108.42	1.137
1200.2	1308.0	107.8	1.090		11	827.27	928.55	101.27	1.122
1238.0	1346.6	108.6	1.088		12	869.28	977.33	108.05	1.124
1326.0	1417.4	91.4	1.069		13	876.77	977.51	100.74	1.115
1417.4	1517.4	100.0	1.071		14	938.60	1035.70	97.10	1.103
1136.44	1236.82	100.38	1.090	Average		794.96	896.56	101.60	1.129
1020.25	1094.88	74.63	1.074	Overall		672.59	748.40	75.81	1.113

†Means of makespan, difference in mean makespan, or ratio of mean makespan for five problems.
‡Means of mean flowtime, difference in mean flowtime, or ratio of mean flowtime for five problems.
§M = # machines; N = # jobs; NSI-SSI = mean difference between NSIST and SSIST makespan or mean flowtime values; NSI/SSI = ratio of mean makespan or mean flowtime for NSIST and SSIST models.

3.4.2 Impact on mean flowtime (Q4)

The corresponding summary data for investigating the impact of separable setup times on mean flowtime is presented in the right half of Table 7. The results for mean flowtime are quite similar to the results for makespan, but with higher average penalties across the board for both M and N. The overall penalty for M = 5, 7, and 9 machines increases, with values of 9.3, 11.7, and 12.9 percent, respectively; and the overall impact is an additional 11.3 percent mean flowtime required per problem. There is again a clear pattern for the NSI/SSI values in each M-value cell, with this ratio steadily declining with increasing values of N. The intercept and slope values for mean flowtime linear regressions are shown in the bottom half of Table 8.

The impact of allowing setup times to be separable from jobs in the regular permutation flowshop has been shown here to be significant, with savings averaging 6.9 percent for makespan and 10.2 percent for mean flowtime values. At the same time, the reader is cautioned that these results are valid only for the processing time and setup time values selected for this initial investigation. We leave to the future a more comprehensive study on the impact of separable setups for a variety of job processing time: setup time ratios, not only for the regular flowshop but for other flowshop variants as well.

Table 8. Regression models for impact of separable setup times

Performance Measure	M‡	Intercept a†	Slope b†
Makespan	5	1.0962	-0.0040
	7	1.1489	-0.0065
	9	1.1434	-0.0053
Mean Flowtime	5	1.1469	-0.0051
	7	1.1765	-0.0056
	9	1.1801	-0.0053

†Estimated values for mean NSI/SSI = a + b[*N*].
‡*M* = # machines; *N* = # jobs.

3.5 Impact of minimizing each performance measure on the other measure

From Conway et al. (1967) onward, flowshop researchers have debated which performance measure is more appropriate, makespan or mean flowtime. (As shown in Eq. (7) and (8), minimizing total flowtime is equivalent to minimizing mean flowtime.) At the same time, few, if any, of these researchers have ever reported the effect minimizing makespan has on the value of mean flowtime for a problem, and *vice versa*. The data collected in both experiments I and II for this paper allow an investigation of these effects, for both the SSIST and NSIST flowshop models.

3.5.1 Impact of makespan on mean flowtime (Q5)

The impact of minimizing makespan on the mean flowtime measure, for the same problems, is shown in the left portion of Table 9. For example, for the $M \times N = 5 \times 6$ cell, the value of 1.124 indicates an average increase of 12.4 percent over the optimal value of mean flowtime for those five problems. Likewise, minimizing makespan increases the mean flowtime for the 5 × 6 cell problems solved as regular flowshop problems (NSIST) by an average of 6.8 percent, compared to the optimal mean flowtime values. For the SSIST model, the overall means for the *M* = 5-, 7-, and 9-machine cells were 11.2, 8.9, and 7.7 percent respectively, thus indicating a downward trend for increasing numbers of machines. For the NSIST model, these values were 9.0, 8.2, and 8.0 percent, respectively, for *M* = 5, 7, and 9 machines. Simple regression analyses of this measure found no statistically significant trends as a function of increasing numbers of jobs, *N*, in any of the *M*-value cells for either the SSIST or NSIST models.

Table 9. Impact of minimizing makespan or mean flowtime on each other

Impact on:		Mean Flowtime			Makespan	
for Minimizing:		Makespan			Mean flowtime	
M§	N§	SSIST	NSIST	Model:	SSIST	NSIST
5	6	1.124†	1.068†		1.076‡	1.075‡
	7	1.084	1.054		1.099	1.090
	8	1.148	1.106		1.091	1.056
	9	1.097	1.079		1.089	1.082
	10	1.096	1.088		1.117	1.117
	11	1.115	1.107		1.062	1.063
	12	1.113	1.092		1.073	1.082
	13	1.104	1.097		1.085	1.088
	14	1.149	1.096		1.090	1.082
	15	1.123	1.117		1.069	1.077
Average		1.112	1.090		1.085	1.081
7	6	1.067	1.080		1.108	1.084
	7	1.076	1.071		1.113	1.101
	8	1.078	1.075		1.089	1.064
	9	1.104	1.103		1.126	1.109
	10	1.071	1.078		1.102	1.067
	11	1.116	1.093		1.110	1.101
	12	1.103	1.062		1.047	1.060
	13	1.085	1.082		1.108	1.094
	14	1.112	1.104		1.110	1.089
	15	1.083	1.071		1.086	1.080
Average		1.089	1.082		1.100	1.085
9	6	1.085	1.054		1.082	1.063
	7	1.090	1.075		1.081	1.075
	8	1.093	1.081		1.059	1.051
	9	1.053	1.047		1.094	1.094
	10	1.038	1.059		1.106	1.078
	11	1.087	1.075		1.101	1.094
	12	1.071	1.055		1.108	1.098
	13	1.087	1.079		1.087	1.091
	14	1.087	1.069		1.113	1.098
Average		1.077	1.066		1.092	1.082
Overall		1.093	1.080		1.092	1.083

†[*Fbar*(for Min *Fmax*)/Min *Fbar*] for SSIST and NSIST models.
‡[*Fmax*(for Min *Fbar*)/Min *Fmax*] for SSIST and NSIST models.
§*M* = # machines; *N* = # jobs; *Fmax* = makespan; *Fbar* = mean flowtime.

3.5.2 Impact of mean flowtime on makespan (Q5)

The impact of minimizing mean flowtime on the makespan measure, for the same set of problems, is shown on the right side of Table 9. For example, for the SSIST model, minimizing mean flowtime for the $M \times N = 5 \times 6$ cell caused an average increase of 7.6 percent over the optimal values of makespan for those five problems. For the SSIST model, the overall increases were 8.5, 10.0, and 9.2 percent respectively for the 5-, 7-, and 9-machine problems For the NSIST model, these same increases were 8.1, 8.5, and 8.3 percent respectively. Again, there was so statistically significant trends within the *M*-value cells for increasing values of *N*, for either model. And there is no apparent trend among the *M*-value cells for either model. Thus, while there is a significant increase above optimal values for either makespan or mean flowtime, as the other measure is minimized, for both the SSIST and NSIST models, there is no detectable trend as a function of increasing numbers of jobs, for the problems solved for this paper.

4. CONCLUSIONS, AND FUTURE RESEARCH SUGGESTIONS

This paper presented an MILP model for the permutation flowshop wherein the setup times were both separable from the job processing times and independent of a job's position in the processing sequence. Two experiments were conducted to estimate the computer times necessary to solve problems with up to 9 machines and 15 jobs, and to then compare these solution time requirements to those required to solve the same sets of problems solved as regular (NSIST) flowshop problems. The resultant data were then used to assess the impact on two optimal sequence performance measures, makespan and mean flowtime when setup times were separated from their jobs and allowed to begin as soon as the machine was free from the preceding job. Lastly, the data were used to analyze the impact on mean flowtime when makespan is minimized, and the impact on makespan when mean flowtime is minimized. We now draw conclusions from this work and then offer some suggestions for future research.

4.1 Conclusions

The MILP model for the SSIST flowshop problem presented in this paper is based on the same JAML constraints approach as that used by Wagner (1959), Stafford (1988), and Tseng and Stafford (2001). We draw the following conclusions regarding this model:

1 The SSIST MILP model extends to *M* machines the optimal sequencing technique of Yoshida and Hitomi (1979) for the 2-machine flowshop problem with separable setup times. Unlike the Yoshida and Hitomi procedure, the SSIST model is not polynomial solvable.

2 The SSIST model is a viable and optimizing model for solving SSIST problems of moderate size. It may be used to optimize SSIST flowshop problems for both the makespan and mean flowtime criteria and for three other criteria not normally found in the flowshop literature.

3 The SSIST model is of comparable problem size complexity with its NSIST counterpart (see Table 2), and the required range of computer solution times for this model, for a given problem size, is comparable to the times required by the NSIST model for the same problem size (see Tables 3, 4, and 5).

4 On average, the SSIST model, using either the makespan or mean flowtime criterion, solves problems faster that its NSIST counterpart, WST, recently tested in the literature (Stafford et al., 2005).

The SSIST model was used to investigate the impact of the separable setup time assumption on both optimal job sequence makespan and optimal sequence mean flowtime. The following conclusions may be drawn regarding the results of this investigation:

5 Separable setup times result in a significant reduction of the optimal sequence makespan relative to the optimal makespan for the flowshop with nonseparable setup times (see Table 7). This reduction increases with increasing numbers of machines in the flowshop, but it decreases slightly with increasing numbers of jobs for a given number of machines.

6 Separable setup times result in a significant reduction of the optimal sequence mean flowtime relative to the optimal mean flowtime for the flowshop with nonseparable setup times. This reduction increases with increasing numbers of machines, but it decreases slightly with increasing numbers of jobs for a given number of machines.

We point out that conclusions 5 and 6 are drawn only for the problem sizes and setup times: job processing times ratios used in this paper. At the same time, we conjecture that similar results will occur for larger problem sizes and other setup times: job processing time ratios as well.

An understanding of the results summarized in Conclusions 5 and 6, which are based mainly on Table 7, is enhanced by considering a JAML diagram of the problem as depicted in Figure 1. As more machines are added to a flowshop (the vertical dimension of Figure 1), there are additional opportunities for reducing the overall processing time of each job since the setups of each job can be performed prior to that job's arrival at each machine. On the other hand, there is essentially no opportunity to decrease makespan on the typical machine *r* (the horizontal dimension of Figure 1) with separable setups because that machine must still experience SS_{rj}, TT_{rj}, $SS_{r,j+1}$, $TT_{r,j+1}$, *et cetera* as the jobs in all *N* positions of the sequence are processed, one after the other on Machine *r*. (See the APPENDIX for an explanation of the *SS* and *TT* terms.) That is, there is no overlap makespan savings between jobs on the same machine. It seems reasonable that this "within jobs" explanation for makespan carries over to mean flowtime as well.

The SSIST model was also used to investigate the impact of minimizing makespan on a job sequence's mean flowtime, and to investigate the impact of minimizing mean flowtime on a job sequence's makespan. We draw the following tentative conclusions regarding these impacts:

7 Minimizing makespan causes an increase in the resultant mean flowtime (over the optimal value) for both the SSIST and NSIST models, and this increase tends to be slightly higher for the SSIST model for a given $M \times N$ problem size (see Table 9).

8 Minimizing mean flowtime causes an increase in the

resultant makespan (over the optimal value) for both the SSIST and NSIST models, with this increase being slightly higher for the SSIST model for a given $M \times N$ problem size.

We caution that the true increases in either makespan or mean flowtime, when minimizing the other measure, most likely are lower than those shown in Table 8. Neither LINDO nor LINGO reports alternate optimal solutions for an integer programming problem, so that, for a given optimal makespan value, there could be an alternate optimal sequence with a different, perhaps lower mean flowtime value. This is especially true for a 15-job problem which has approximately 1.31×10^{12} different sequences to investigate.

4.2 Suggestions for future research

Based on our development and investigation of the MILP model for the flowshop with separable, sequence-independent setup times, we enumerate three of several possible projects for future investigation. First, it is possible that the techniques detailed by Allahverdi (2000) may be combinable with the current SSIST model to improve computer solution times for the mean flowtime criterion. Second, following Stafford et al. (2005), there may be alternate SSIST MILP models which yield optimal solution in shorter times than experienced with the current SSIST model. Third, it would be useful to run an extensive study on the influences of makespan and mean flowtime on each other, not only for the SSIST model, but for the regular flowshop environment as well.

ACKNOWLEDGMENT

The authors wish to acknowledge the valuable inputs and suggestions of three anonymous referees. These inputs increased the readability and accuracy of this paper significantly. Any remaining inaccuracies are the responsibility of the authors.

REFERENCES

1. Aggarwal, S.C. and Stafford, E.F. (1975). A heuristic algorithm for the flowshop problem with a common job sequence on all machines. *Decision Sciences*, 6: 237-251.
2. Aldowaisan, T. (2001). A new heuristic and dominance relations for no-wait flowshops with setups. *Computers and Operations Research*, 28: 563-584.
3. Aldowaisan, T. and Allahverdi, A. (1998). Total flowtime in no-wait flowshops with separated setup times. *Computers and Operations Research*, 25: 757-765.
4. Allahverdi, A. (2000). Minimizing mean flowtime in a two-machine flowshop with sequence-independent setup times. *Computers and Operations Research*, 27: 111-127.
5. Allahverdi, A., Gupta, J.N.D., and Aldowaisan, T. (1999). A review of scheduling research involving setup considerations. *OMEGA*: *The International Journal of Management Science*, 27: 219-239.
6. Baker, K.R. (1975). An elimination method for the flow-shop problem. *Operations Research*, 23: 159-162.
7. Chen, B., Potts, C.N., and Woeginger, G.J. (1998). A review of machine scheduling: complexity, algorithms and applications. In: Du, D.Z. and Du Pardalos, P.M. (Eds.), *Handbook of Combinatorial Optimization*, Kluwer, Dordrecht, Netherlands, pp.21-169.
8. Conway, R.W., Maxwell, W.L., and Miller, L.W. (1967). *Theory of Scheduling*, Addison-Wesley, Reading, Massachusetts.
9. Corwin, B.D. and Esogbue, A.O. (1974). Two-machine flowshop scheduling problems with sequence dependent setup times: a dynamic programming approach. *Naval Research Logistics Quarterly*, 21: 515-524.
10. Graves, S.C. (1981). A review of production scheduling. *Management Science*, 29: 646-675.
11. Gupta, J.N.D. (1972). Heuristic algorithms for multistage flowshop scheduling problem. *AIIE Transactions*, 4: 11-18.
12. Gupta, J.N.D. (1975). Analysis of combinatorial approach to flowshop scheduling problems. *Journal of the Operational Research Society*, 26: 431-440.
13. Gupta, J.N.D. and Darrow, W.P. (1986). The two-machine sequence dependent flowshop scheduling problem. *European Journal of Operational Research*, 24: 439-446.
14. Gupta, J.N.D. and Stafford, E.F. (2006). Flowshop scheduling research after five decades. *European Journal of Operational Research*, 169: 699-711.
15. Hall, N.G. and Sriskandarajah, C. (1996). A survey of machine scheduling problems with blocking and no-wait in process. *Operations Research*, 44: 510-525.
16. Ignall, E. and Schrage, L. (1965). Application of the branch-and-bound technique to some flow-shop scheduling problems. *Operations Research*, 13: 400-412.
17. Johnson, S.M. (1954). Optimal two- and three-stage production schedules with set-up times included. *Naval Research Logistics Quarterly*, 1: 61-68.
18. Kang, B.S. and Markland, R.E. (1988). Evaluation of scheduling techniques for solving flowshop problems with no intermediate storage. *Journal of Operations Management*, 7: 1-24.
19. Lageweg, B.J. (1978). A general bounding scheme for the permutation flow-shop problem. *Operations Research*, 26: 53-67.
20. Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G., and Shmoys, D.B. (1993). Sequencing and scheduling: algorithms and complexity. In: Graves, S.C., Rinnooy Kan, A.H.G., and Zipkin, P.H. (Eds.), *Logistics of Production and Inventory*, North Holland, Amsterdam, Netherlands, pp. 445-522.
21. LINDO Systems, Inc. (1999). *LINDO Users Manual*, LINDO Systems Inc., Chicago, Illinois.
22. LINDO Systems, Inc (2004). *LINGO User's Guide*, LINDO Systems Inc, Chicago.
23. Manne, A.S. (1960). On the job-shop scheduling problem. *Operations Research*, 8: 219-223.
24. Norman, B.A. (1999). Scheduling flowshops with finite buffers and sequence-dependent setup times. *Computers*

and Industrial Engineering, 36: 163-177.
25. Papadimitriou, C.H. and Kanellakis, P.C. (1980). Flowshop scheduling with limited temporary storage. *Journal of the Association for Computing Machinery*, 27: 533-54.
26. Rios-Mercado, R.Z. and Bard, J.F. (1998). Computational experience with a branch-and-cut algorithm for flowshop scheduling with setups. *Computers and Operations Research*, 25: 351-366.
27. Srikar, B.N. and Ghosh, S. (1986). A MILP model for the *N*-job *M*-stage flowshop with sequence dependent set-up times. *International Journal of Production Research*, 24: 1459-1474.
28. Stafford, E.F. (1983). A mixed-integer linear programming model for the classical flowshop sequencing problem. Working paper, University of South Carolina.
29. Stafford, E.F. (1988). On the development of a mixed-integer linear programming model for the standard flowshop. *Journal of the Operational Research Society*, 39: 1163-1174.
30. Stafford, E.F. and Tseng, F.T. (1990). On the srikar-ghosh MILP model for the *N* × *M* SDST flowshop problem. *International Journal of Production Research*, 28: 1817-1830.
31. Stafford, E.F. and Tseng, F.T. (2002). Two models for a family of flowshop sequencing problems. *European Journal of Operational Research*, 142: 282-293.
32. Stafford, E.F., Tseng, F.T., and Gupta, J.N.D. (2005). Comparative evaluation of MILP flowshop models. *Journal of the Operational Research Society*, 56: 88-101.
33. Tseng, F.T. and Stafford, E.F., (2001). Two MILP models for the *N* × *M* SDST flowshop sequencing problem. *International Journal of Production Research*, 39: 1777-1809.
34. Tseng F.T., Stafford, E.F., and Gupta, J.N.D. (2004). An empirical analysis of integer programming formulations for the permutation flowshop. *OMEGA*: *The International Journal of Management Science*, 32:285-293.
35. Wagner, H.M. (1959). An integer linear-programming model for machine scheduling. *Naval Research Logistics Quarterly*, 6: 131-140.
36. Wismer, D.A. (1972). Solution of the flowshop-scheduling problem with no intermediate queues. *Operations Research*, 20: 687-697.
37. Yoshida, T. and Hitomi, K. (1979). Optimal two-stage production scheduling with setup times separated. *AIIE Transactions*, 11: 261-263.

APPENDIX: DERIVATION OF EQUATIONS 3 AND 4 FOR THE SSIST MODELS

We first define the following four sets of "model development" variables which will facilitate the understanding of the development of Eq. (3) and (4) for the SSIST/Reg model. Let: $\boldsymbol{B} = \{B_{rj}\}$, where B_{rj} = beginning time of the job in position j of the sequence on machine r; $\boldsymbol{C} = \{C_{rj}\}$, where C_{rj} = completion time of the job in position j of the sequence on machine r; $\boldsymbol{SS} = \{SS_{rj}\}$, where SS_{rj} = setup time of the job in position j on machine r; and $\boldsymbol{TT} = \{TT_{rj}\}$, where TT_{rj} = processing time on machine r for the job in position j. These variables disappear from the final form of Eq. (3) and (4) as shown below.

In terms of the decision variables and system parameters defined in Section 2, SS_{rj} and TT_{rj} may be expressed as follows:

$$SS_{rj} = \sum_{i=1}^{N} S_{ri} Z_{ij} \quad \{j = 1, ..., N; r = 1, ..., M\} \tag{A-1}$$

$$TT_{rj} = \sum_{i=1}^{N} P_{ri} Z_{ij} \quad \{j = 1, ..., N; r = 1, ..., M\} \tag{A-2}$$

Two types of constraints are first developed and then combined into a set of equations.: (1) job adjacency constraints, in which, on each machine, job i must not be started until its predecessor has been completely processed and the machine has been set up to process job i; and (2) machine linkage constraints, in which a job i cannot be started on a machine j until that job has finished being processed on machine $j - 1$ and machine j has been set up to process job i. In other words, a feasible solution requires the satisfaction of the following:

1 Each machine r must be set up before a job i can be processed on it. Machine r may be set up for job i as long as job i's predecessor job has finished processing on machine r, even if job i is not ready to be processed on machine r (that is, job i is still being processed on machine $r - 1$).

2 The job in position j must completely precede the job in position $j + 1$ for all M machines. That is, the beginning time of the job in position $j + 1$ must not be earlier than the completion time of the job in position j plus the setup time of changing over from the job in position j to the job in position $j + 1$, on machine r. This requirement may be expressed as:

$$C_{rj} + SS_{r,j+1} \le B_{r,j+1} \tag{A-3}$$

3 Each job must be completed on machine r before it can begin processing on machine $r + 1$. Thus the beginning time of the job in position j on machine $r + 1$ must not be earlier than the completion time of that same job on machine r. This may be expressed as:

$$C_{rj} \le B_{r+1,j} \tag{A-4}$$

4 A job cannot be split; once a job has started on a machine it cannot be interrupted until it is completed. Thus the completion time of the job in position j on machine r equals its processing start time plus its processing time on machine r, as shown in the following:

$$C_{rj} = B_{rj} + TT_{rj} \quad \text{(A-5)}$$

Eq. (A-3) through (A-5) hold for each job in position j on machine r ($j = 1, ..., N-1$; $r = 1, ..., M-1$).

In Eq. (A-3), the difference between the two sides of the inequality is the possible idle time of machine r, $X_{r,j+1}$, before it starts processing the job in position $j + 1$. Thus Eq. (A-3) may be rewritten as:

$$C_{rj} + SS_{r,j+1} + X_{r,j+1} = B_{r,j+1} \quad \text{(A-6)}$$

For Eq. (A-4), the difference between the two sides of the inequality is the possible idle time of job j, Y_{rj}, after that job finishes processing on machine r and before it begins its processing on machine $r + 1$. Eq. (A-4) may then be rewritten as:

$$C_{rj} + Y_{rj} = B_{r+1,j} \quad \text{(A-7)}$$

The processing time for the job in position $j + 1$ on machine r, TT_{rj+1}, may be added to both sides of Eq. (A-6) yielding:

$$C_{rj} + SS_{r,j+1} + X_{r,j+1} + TT_{r,j+1} = B_{r,j+1} + TT_{r,j+1} \quad \text{(A-8)}$$

But, from Eq. (A-5), the right hand side of Eq. (A-8) is just the completion time of the job in position $j + 1$ on machine r, so Eq. (A-8) may be rewritten as:

$$C_{rj} + SS_{r,j+1} + X_{r,j+1} + TT_{r,j+1} = C_{r,j+1} \quad \text{(A-9)}$$

Likewise, adding $TT_{r+1,j}$ to both sides of Eq. (A-7), then comparing its right hand side to Eq. (A-5), yields the following:

$$C_{rj} + Y_{rj} + TT_{r+1,j} = C_{r+1,j} \quad \text{(A-10)}$$

Eq. (A-9) represents the usage of machine r between the completion of the job in position j on that machine and the completion of the job in position $j + 1$ on that machine. Upon completion on machine r, the job in position j is transferred to machine $r + 1$. Machine r is setup for the job in position $j + 1$. Then, after a possible machine delay waiting for the job in position $j + 1$, $X_{r,j+1}$, machine r will process that next job, $TT_{r,j+1}$.

Eq. (A-10) represents the job in position j between completion of processing on two successive machines. Upon transfer to machine $r + 1$, the job in position j may experience a delay, Y_{rj} before it can begin processing on machine $r + 1$. Then this job must be processed on machine $r + 1$, $TT_{r+1,j}$, to reach its completion on that machine.

Next, consider Eq. (A-7) for the job in position $j + 1$ which results in:

$$C_{r,j+1} + Y_{r,j+1} = B_{r+1,j+1} \quad \text{(A-11)}$$

and Eq. (A-6) for machine $r + 1$, which results in:

$$C_{r+1,j} + SS_{r+1,j+1} + X_{r+1,j+1} = B_{r+1,j+1} \quad \text{(A-12)}$$

The earliest time the job in position $j + 1$ can start processing on machine $r + 1$ is the maximum of: (1) completion time of itself (job $j + 1$) on machine r, $C_{r,j+1}$; and (2) the time when machine $r + 1$ is first available to begin processing job $j + 1$, ($C_{r+1,j} + SS_{r+1,j+1}$). If the completion time in condition 1 is earlier, then the job will have some waiting time, $Y_{r,j+1}$, as shown in Eq. (A-11). If the completion time in condition 2 is earlier, then machine $r + 1$ will have waiting time, $X_{r+1,j+1}$, as shown in Eq. (A-12).

Replacing the $C_{r,j+1}$ term in Eq. (A-11) with Eq. (A-9) results in:

$$C_{rj} + SS_{r,j+1} + X_{r,j+1} + TT_{r,j+1} + Y_{r,j+1} = B_{r+1,j+1} \quad \text{(A-13)}$$

Likewise, replacing the $C_{r+1,j}$ term in Eq. (A-12) with Eq. (A-10) yields:

$$C_{rj} + Y_{rj} + TT_{r+1,j} + SS_{r+1,j+1} + X_{r+1,j+1} = B_{r+1,j+1} \quad \text{(A-14)}$$

Since Eq. (A-13) and (A-14) both express the difference between the time the job in position $j + 1$ starts processing on machine $r + 1$ and the time the job in position j finished its processing on machine r, we may set these equations equal to each other, yielding:

$$SS_{r,j+1} + X_{r,j+1} + TT_{r,j+1} + Y_{r,j+1} = Y_{rj} + TT_{r+1,j} + SS_{r+1,j+1} + X_{r+1,j+1} \quad \text{(A-15)}$$

The tight binding between each pair of adjacent jobs and each pair of consecutive machines represented by Eq. (A-15) is depicted in Figure A1.

Eq. (A-15) may be rewritten as follows:

$$(SS_{r,j+1} - SS_{r+1,j+1}) + (TT_{r,j+1} - TT_{r+1,j}) + (X_{r,j+1} - X_{r+1,j+1}) + (Y_{r,j+1} - Y_{rj}) = 0 \quad \text{(A-16)}$$

Finally, Eq. (A-1) and (A-2) are used to replace the SS and TT terms of Eq. (A-16), and when like terms are combined, the result is the constraints represented by Eq. (3):

$$\sum_{i=1}^{N} (S_{ri} + P_{ri} - S_{r+1,i}) Z_{i,j+1} - \sum_{i=1}^{N} P_{r+1,i} Z_{ij} + (X_{r,j+1} - X_{r+1,j+1}) + (Y_{r,j+1} - Y_{rj}) = 0$$
$$\{j = 1, ..., N-1;\ r = 1, ..., M-1\} \quad (3)$$

Effectively, several of the constraints described above are combined into Eq. (3).

Now consider the development of Eq. (4) which deals with the first job in the processing sequence. The earliest starting time for the first job in the sequence is immediately after the first machine has been set up. The starting time of this first job on subsequent machines, 2 through M, is determined by the maximum of (1) the completion time of this job on the previous machine, $r-1$; and (2) the setup time on the current machine r. If the completion time on the previous machine occurs first, then this first job has idle time $Y_{r-1,1}$ before it can proceed on the current machine r. Otherwise, machine r has idle time X_{r1} while awaiting the first job to be completed on the previous machine.

Using Figure A2, the following expression may be written to account for parallel segments of the Gantt chart for the start of the first job in the sequence on each of the M machines:

$$SS_{r1} + X_{r1} + TT_{r1} + Y_{r1} = SS_{r+1,\,1} + X_{r+1,\,1}$$
$$\{r = 1, ..., M-1\} \qquad \text{(A-17)}$$

Eq. (A-17) equals the start time of the job in the first position in the sequence for machines 2 to M.

Eq. (A-1) may be substituted for the SS terms in Eq. (A-17), and Eq. (A-2) may be substituted for the TT terms. Rearranging and collecting similar terms results in Eq. (4):

$$\sum_{i=1}^{N}(S_{ri} + P_{ri} - S_{r+1,\,i})Z_{i1} + (X_{r1} - X_{r+1,\,1}) + Y_{r1} = 0.$$
$$\{r = 1, ..., M-1\} \qquad (4)$$

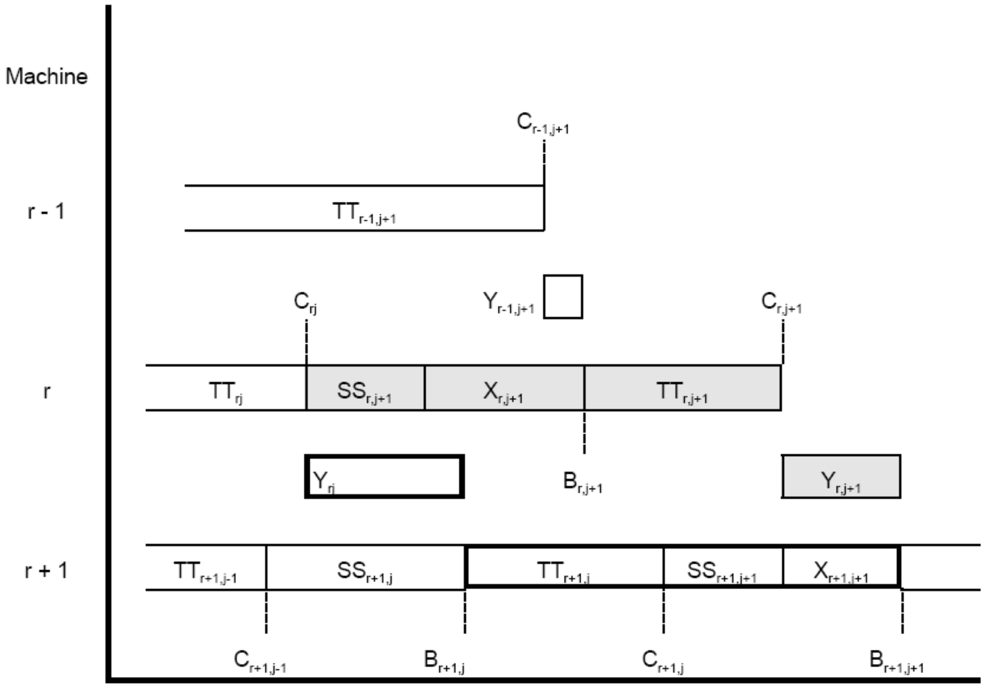

Figure A1. Relationship between adjacent jobs and machines.

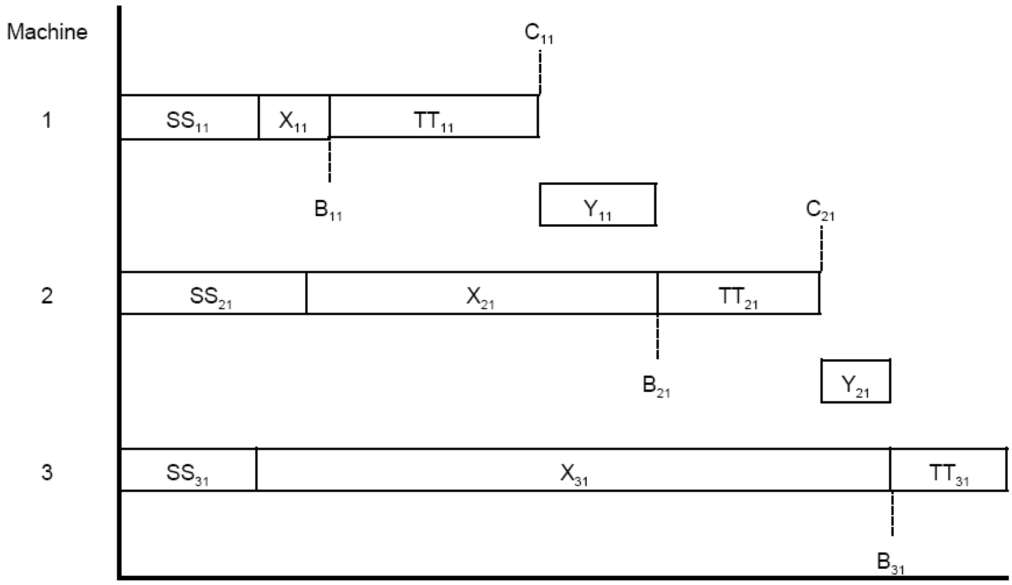

Figure A2. Variables depicting the start of the job in the first sequence position.