Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Omega 32 (2004) 285–293
omega
The International Journal of Management Science
www.elsevier.com/locate/dsw

An empirical analysis of integer programming formulations for the permutation flowshop

Fan T. Tseng, Edward F. Stafford Jr., Jatinder N.D. Gupta*

College of Administrative Science, The University of Alabama in Huntsville, Huntsville, AL 35899, USA

Received 25 September 2002; accepted 2 December 2003

Abstract

An empirical analysis was conducted to assess the relative effectiveness of four integer programming models for the regular permutation flowshop problem. Each of these models was used to solve a set of 60 flowshop problems. Analysis of the resultant computer solution times for each model indicated that the two assignment problem based models solved these problem instances in significantly less computer time than either of the two dichotomous constraints based models. Further, these computer solution time differences increased dramatically with increased numbers of jobs and machines in the flowshop problem. These results contradict Pan's conclusion that a variant of Manne's dichotomous constraints approach was superior to the assignment problem approaches of Wagner and Wilson because the Manne model required less than half of the binary integer variables required by the assignment problem based models.

Keywords: Flowshop scheduling; Minimizing makespan; Mixed-integer linear programming; Computational comparisons

1. Introduction

Wagner [1] proposed all-integer mathematical programming formulations for the general job shop scheduling problem and for the regular flowshop scheduling problem with permutation schedules. Following Wagner's seminal work, several authors have proposed alternative mathematical formulations for these scheduling problems. Bowman [2], Manne [3], Gupta [4], and Morten and Pentico [5] proposed alternative approaches for the general job shop problem. Baker [6], Stafford [7], and Wilson [8] described mixed-integer linear programming (MILP) approaches to minimize makespan in a regular flowshop with permutation schedules. These models can be divided into two broad classes depending on the type of constraints used to assign jobs to various sequence positions. The *Manne models* use pairs of *dichotomous constraints*, or their algebraic equivalents, to control the assignment of jobs to various sequence positions. The *Wagner* and *Wilson models*, thought to be the best formulation for the permutation flow-shop problem, use the classic *assignment problem* for the same assignment task. However, in comparing the number of binary integer variables required in each alternative integer programming model for various scheduling problems, Pan [9] concludes that for the permutation flowshop problems: [1]

a. Manne's model is the best MILP formulation.
b. Wagner's model is second best MILP formulation.
c. Wilson's model trails behind Wagner's formulation.

While the number of binary variables is an important aspect of the MILP problem size complexity, numbers of constraints needed and the total matrix size generated can also affect the computational effort required to solve the problem.

* Corresponding author. Tel.: +1-256-824-6593; fax: +1-256-824-2929.
E-mail address: guptaj@uah.edu (J.N.D. Gupta).

[1] Pan's conclusions also deal with job-shop and regular flow-shop problems *without* permutation schedules; however, since these problems are not considered in this paper, only the conclusions related to permutation flowshops are reported here.

0305-0483/$ - see front matter
doi:10.1016/j.omega.2003.12.001

Therefore, the above MILP model ranking proposed by Pan [9] may well require some reconsideration. One way to do this would be to empirically evaluate the computational effort required to solve the same set of permutation flowshop problems using various MILP models. However, the computational studies of these MILP models for the permutation flowshops is very limited. Although Wilson [8] did solve a small set of common problems with his model and with a variant of the Wagner model, his model has otherwise been untested. The Manne type models evaluated in this paper have never been fully tested in a formal experimental design. Further, Pan's model (which is actually based on a job shop model of Liao and You [10] rather than the original Manne formulation), has never been tested. For several related flowshop problems, two recent papers suggest that assignment problem based Wagner type MILP models outperform dichotomous constraints based Manne type models with regard to computer solution times when makespan is minimized [11,12].

The purpose of this paper is to empirically investigate, for the first time on a common set of problems, the computer solution time requirements for competing MILP models for minimizing makespan in a flowshop problem with permutation schedules (hereafter referred to simply as the *flowshop problem*). This investigation also enables us to correct and reassess Pan's ranking of various MILP models for solving the permutation flowshop problems. Specifically, this paper compares two Manne-type MILP models to the Wagner and Wilson MILP models for the flowshop problem. Since all four models provide optimal solutions to the problems solved, the basis of comparison is the amount of computer time (measured by the CPU time) each model requires to find the optimal solutions for a set of common problems.

The rest of the paper is organized as follows. Section 2 briefly presents the four integer programming models used in this study. In doing so, this section also describes the corrections needed to Pan's MILP formulation of the flowshop problem. Section 3 provides an evaluation of these models in terms of size complexity and computer solution times (measured by CPU time) for a common set of 60 problems. Finally, Section 4 concludes the paper and suggests some fruitful directions for future research in this area.

2. The regular flowshop models

Since Johnson's [13] seminal paper, the regular permutation flowshop scheduling problem has constituted an important and growing topic in the production and operations research literature. This problem has two major elements: (1) a set of N jobs; and (2) a production system of M machines on which these N jobs are to be processed. Each of these N jobs has the same order of processing on the M machines. The objective of this problem is to find that schedule (out of the $N!$ possible sequences) which minimizes makespan. The standard assumptions of the regular permutation flowshop problem have been well documented by Gupta [14] and others. Therefore, they are not repeated here.

2.1. Model notations

The subscript symbols are: r for machines, $(1 \leqslant r \leqslant M)$; i and k for jobs, $(1 \leqslant i, k \leqslant N)$; and j for the sequence position, $(1 \leqslant j \leqslant N)$ where the parameters M and N represent the numbers of machines and jobs, respectively. $T = \{T_{rj}\}$ is the $M \times N$ matrix of job processing times, with T_{ri} = processing time of job i on machine r. The variables are defined as follows:

- B_{rj} start (begin) time of job in sequence position j on machine r
- C_{ri} completion time of job i on machine r
- D_{ik} 1, if job i is scheduled any time before job k; 0, otherwise; $i < k$
- E_{rj} completion (end) time of the job in position j on machine r
- S_{ri} start time of job i on machine r
- 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
- $C_{\max}$ maximum flowtime (makespan) of the schedule determined by the completion time of the job in the last sequence position on the last machine.

D_{ik} and Z_{ij} are binary integer variables. The others are real variables that take integer values when processing times are also given integer values. Hence all models described below are MILP models.

2.2. The Wagner model

Wagner [1] formulated an all-integer programming model for a three-machine flowshop. Baker [6] and Stafford [7] extended this all-integer model to accommodate problems with $M > 3$. Stafford also converted the all-integer model to an MILP model and added three sets of constraints (omitted by both Wagner and Baker) to insure that the Gantt chart representation was "anchored" to the zero time line. The corrected and simplified "Wagner" MILP model for the permutation flowshop is as follows:

Minimize $\quad C_{\max} = C_{MN}$

subject to

$$\sum_{j=1}^{N} Z_{ij} = 1; \quad (1 \leqslant i \leqslant N), \tag{1}$$

$$\sum_{i=1}^{N} Z_{ij} = 1; \quad (1 \leqslant j \leqslant N), \tag{2}$$

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

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

$$C_{MN} = \sum_{i=1}^{N} T_{Mi} + \sum_{p=1}^{N} X_{Mp}. \tag{5}$$

Constraints (1) and (2) are the classical assignment problem, with (1) insuring that each job is assigned to just one sequence position, while (2) insures that each sequence position is filled with only one job. Constraints (3) and (4), termed job-adjacency, machine-linkage (JAML) constraints by Tseng and Stafford [11], insure that (a) the job in sequence position j cannot begin processing on machine $r+1$ until it has completed its processing on machine r; and (b) the job in sequence position $j+1$ cannot begin its processing on machine r until the job in sequence position j has completed its processing on that same machine. (The formulations of both Wagner and Baker omitted constraint (4).) Based on an earlier paper by Liao and You [10], Pan [9] also made the claim that constraint (4) or its equivalent is redundant, and thus not needed in the formulation. Stafford and Tseng [15] recently proved this claim to be incorrect. Constraint (5) measures makespan of the set of jobs.

2.3. The Wilson model

Instead of using sets of equality constraints to control the JAML relationships, Wilson [8] used sets of inequality constraints similar to those used in the Manne models described later in this paper. Wilson's model is formulated as follows:

$$\text{Minimize} \quad C_{\max} = B_{MN} + \sum_{i=1}^{N} T_{Mi} Z_{iN}$$

subject to

$$\sum_{j=1}^{N} Z_{ij} = 1; \quad (1 \leqslant i \leqslant N), \tag{6}$$

$$\sum_{i=1}^{N} Z_{ij} = 1; \quad (1 \leqslant j \leqslant N), \tag{7}$$

$$B_{1j} + \sum_{i=1}^{N} T_{1i} Z_{ij} = B_{1,j+1}; \quad (1 \leqslant j \leqslant N-1), \tag{8}$$

$$B_{11} = 0, \tag{9}$$

$$B_{r1} + \sum_{i=1}^{N} T_{ri} Z_{i1} = B_{r+1,1}; \quad (1 \leqslant r \leqslant M-1), \tag{10}$$

$$B_{rj} + \sum_{i=1}^{N} T_{ri} Z_{ij} \leqslant B_{r+1,j}; \quad (1 \leqslant r \leqslant M-1; 2 \leqslant j \leqslant N), \tag{11}$$

$$B_{rj} + \sum_{i=1}^{N} T_{ri} Z_{ij} \leqslant B_{r,j+1}; \quad (2 \leqslant r \leqslant M; 1 \leqslant j \leqslant N-1). \tag{12}$$

Constraints (6) and (7) represent the assignment problem described in the Wagner model above. Constraints (8), (9), and (10) insure there is no idle time on machine 1, and that job 1 processes on all M machines without delay. Constraint (11) insures that the start of each job on machine $r+1$ is no earlier than its finish on machine r. Constraint (12) insures that the job in position $j+1$ in the sequence does not start on machine r until the job in position j in the sequence has completed its processing on that machine.

2.4. The Manne model

Manne [3] proposed a dichotomous-constraints integer programming model for the general job shop problem. In his model, Manne assured that only one of each pair of constraints can hold, so that job i either precedes job k somewhere in the processing sequence, or it does not, thus implying that job k precedes job i. Following Stafford and Tseng [16], this true Manne model adapted to a permutation flowshop, with P defined as a very large constant, is as follows:

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

subject to

$$C_{1i} \geqslant T_{1i}; \quad (1 \leqslant i \leqslant N), \tag{13}$$

$$C_{ri} - C_{r-1,i} \geqslant T_{ri}; \quad (2 \leqslant r \leqslant M; 1 \leqslant i \leqslant N), \tag{14}$$

$$C_{ri} - C_{rk} + PD_{ik} \geqslant T_{ri}; \quad (1 \leqslant r \leqslant M; 1 \leqslant i < k \leqslant N), \tag{15}$$

$$C_{ri} - C_{rk} + PD_{ik} \leqslant P - T_{rk}; \quad (1 \leqslant r \leqslant M; 1 \leqslant i < k \leqslant N), \tag{16}$$

$$C_{\max} \geqslant C_{Mi}; \quad (1 \leqslant i \leqslant N). \tag{17}$$

Constraint (13) insures that the completion time of each job on machine 1 occurs no earlier than the duration of that job's processing time on machine 1. Constraint (14) insures that each job's completion time on machine r is no earlier than that job's completion time on machine $r-1$ plus that job's processing time on machine r. Constraints (15) and (16) are the paired *disjunctive* constraints which insure that job i either precedes job k or follows job k in the sequence, but not both. Constraint (17) equates makespan to the maximum completion time of all jobs on the last machine.

2.5. The Liao–You model

In their work, Liao and You [10] algebraically combined each pair of inequality dichotomous constraints from the original Manne [3] model into a single equality constraint that they set equal to a surplus variable, q_{rik}, which accounts for the precedence relationship of jobs i and k on machine r. Again, P is a very large constant. In order to ensure feasibility, they added a boundary constraint on the surplus variables q_{rik}. This modified Manne-like model, which is here labeled the "Liao–You" model, is as follows:

Minimize $\quad C_{\max}$

subject to

$$S_{ri} + T_{ri} \leqslant S_{r+1,i}; \quad (1 \leqslant r \leqslant M-1; 1 \leqslant i \leqslant N), \tag{18}$$

$$S_{ri} - S_{rk} + PD_{ik} - T_{rk} = q_{rik}; \quad (1 \leqslant r \leqslant M; 1 \leqslant i < k \leqslant N), \tag{19}$$

$$P - T_{ri} - T_{rk} \geqslant q_{rik}; \quad (1 \leqslant r \leqslant M; 1 \leqslant i < k \leqslant N), \tag{20}$$

$$C_{\max} \geqslant S_{Mi} + T_{Mi}; \quad (1 \leqslant i \leqslant N). \tag{21}$$

The above permutation flowshop MILP model labeled as the Manne model by Pan [9] is not a true dichotomous constraints based model. Rather, it is a model Pan adapted for the permutation flowshop from the job-shop model of Liao and You [10].

3. Evaluation of the models

Throughout the flowshop literature, competing techniques for problem solution are compared on two primary dimensions: (i) how close to the optimal solution a technique's solution is; and (ii) how much computer solution time a technique requires to reach its best solution. For the current study, all MILP models provide optimal solutions for each problem solved. Hence, computer solution time is the single appropriate measure of model superiority when comparing competing MILP models. However, computational effort needed may depend on the problem size complexity of the MILP formulation of the flowshop problems. Therefore, four different MILP models for the permutation flowshop are evaluated in two different ways. First, the problem size complexities of the models are presented and compared. Second, each model is used to solve a 12-cell experimental design of 60 problems, and the computer solution times required by each model are compared. The results from the experiment are also used to assess the validity of Pan's conclusion regarding the superiority of the Manne model relative to the Wagner and Wilson models.

Table 1
Size complexity of the MILP models

Model	Binary variables[a]	Constraints[a]	Continuous variables[a]
Wagner	N^2	$MN+N+1$	$2MN-N+1$
Wilson	N^2	$2MN-M+N+1$	MN
Manne	$N(N-1)/2$	MN^2+N	$MN+1$
Liao–You	$N(N-1)/2$	MN^2	$MN(N+1)/2+1$

[a] N = number of jobs, M = number of machines.

3.1. Size complexity of the models

Size complexity indicates how large a problem is in terms of binary variables, constraints, and continuous (real) variables as a function of M and N, the number of machines and jobs, respectively, in the problem. The size complexity of each of the four MILP models for the regular permutation flowshop is presented in Table 1.

Some of the results in Table 1 differ from those in Table 3 of Pan [9] in the number of constraints and continuous variables. First, Pan stated that the Wagner model requires $MN-M+N+2$ constraints and $2MN-M-2N+2$ continuous variables which differ significantly from those shown in Table 1 of the current paper. A careful evaluation of Pan's presentation of the Wagner model indicates that his version of the Wagner model does not include the constraints (4) shown above. As mentioned earlier, this type of constraints is required to insure feasibility of the optimal solution for a problem. Without (4), the job in the first sequence position could begin processing on machines 2 through M before it has completed its processing on machine 1 [7,15]. Correcting these errors and removing the unnecessary constraints in Pan's description of the Wagner model shows that the numbers of constraints and continuous variables in his paper will be the same as those reported in the current paper. Second, Pan reported that the number of constraints for the Liao–You model (which he calls the Manne model) is $MN(N+1)/2$. However, if we correctly count the constraints from his formulation, we find that the total number of constraints is MN^2 which is the number reported in Table 1.

The Wagner and Wilson models both rely on an assignment problem approach for selecting the optimal sequence of jobs. Both models require N^2 binary variables. The Wilson model uses $N(M-1)+1$ fewer continuous variables, but at a cost of an additional $M(N-1)$ constraints. The impact of this tradeoff is seen in the computer solution times presented below.

The Manne and Liao–You models both rely on a variation of the dichotomous constraints approach to identify the relative position of jobs in the optimal sequence. Both models require $N(N-1)/2$ binary variables, a number less than half that required by the Wagner and Wilson models. The Wagner model requires more continuous variables than the Manne model. Based solely on numbers of required binary variables, Pan [9] made his claim that the Manne model (Liao–You) is best for the permutation flowshop, and that the Wagner and the Wilson models rank two and three, respectively. While we agree with Pan's analyses regarding problem sizes generated by the competing integer programming models, we disagree with him regarding the *number of binary variables* as the single correct measure of model superiority. There are at least two additional factors that Pan overlooked.

First, the number of constraints, and therefore the rank of the matrix, is also an important factor in determining the computer solution time. For example, the number of constraints for the Wagner model is $MN+N+1$, whereas the number of constraints for the Manne model is MN^2+N, which is *at least* $N(1-1/M)$ *times* that of the Wagner model. It is well known that the computational time for a pivot operation in the simplex method is in the order of the square of the rank of the matrix, which is determined mainly by the number of constraints. Similar comparisons may be made between other combinations of the two assignment problem based models versus the two Manne models.

Second, the size of the matrix as measured by the number of entries in the matrix also affects computational time. The number of entries can be calculated by multiplying the number of constraints by the total number of binary and continuous variables. For the problem sizes studied in this research, the matrix size for the Manne model is between two and four times that of the Wagner model. The ratio of the matrix sizes between the Liao–You and the Wilson models is even larger, between 4 and 12.

3.2. Computer solution times

While the Manne and Liao–You models have less than half of the number of binary variables required by the Wagner and Wilson models, the number of constraints and the matrix size in these two models are much larger than that of the Wagner and Wilson models. This tradeoff could significantly affect the computational time of these models. To investigate this, an experimental design was constructed and executed to examine the relative effectiveness of the four MILP models for the flowshop problem. The performance measure is the computer solution (CPU) time required to solve each problem. Sixty (60) regular flowshop problems were solved by each of the four models described above. These problems were divided into five replications each for the 12 experimental cells represented by $M=5$, 7, and 9 machines; and by $N=6$, 7, 8, and 9 jobs. The job processing time parameters (T_{ri}) were randomly generated from a uniform distribution ranging from 1 to 100.

Four different *model generators* were used to convert the T_{ri} parameters to 240 data sets. Each data set was then solved using the integer programming, terse, and the *Take Command* options of Hyper LINDO 6.1 on a Dell Pentium III 800 MHz personal computer equipped with RDRAM. The LINDO elapsed time option was used to measure actual CPU times to the nearest 0.01 s. Relevant solution time parameters for all problems were recorded in separate text files for each problem solved, and were subsequently summarized in a workbook of large Excel spreadsheets. The conservative, small sample, non-parametric sign test [17] was used to test for significant differences in required computational time for each pair of models.

A summary of the resultant computer solution times for all MILP models is shown in Tables 2 and 3. This summary includes the mean solution time, the standard deviation of the solution times, and the median solution time for each model for each of the 12 cells in the experiment. Additional information included in this summary is discussed below.

3.2.1. Analysis of the means

Following past practice in the literature (see for example [12,18]), the main analysis of the experimental results concentrates on the analysis of the mean computer solution times.

Assignment problem versus dichotomous constraints based models. Both assignment problem based models—Wagner and Wilson—*always required less computer time* than either of the dichotomous constraints based models *for solving all* 60 *flowshop problems*. Further, the differences in required computer solution times increased dramatically with both increasing N and M. The sign test results show that both the Wagner and Wilson models did use statistically significantly less time than either the Manne or the Liao–You models *for each of the* 12 *cells* in this experiment ($p=0.03125$).

Assignment problem based problem models: *Wagner versus Wilson*. The values in the second to the last column in Table 3 indicate that the Wagner model found the optimal solution in less time than the Wilson model in 52 of the 60 problems solved for this paper. When using the sign test for a cell size of 5, the test statistic X, the number of times one model finds the optimal solution in less computer time than the other model, must equal 5 for the test to be statistically significant. Thus, for half of the 12 cells in this experiment, there is no statistical difference between the two assignment problem based models. On the other hand, when the cells are combined by values of M (new cell size $=20$), the Wagner model does use significantly less time than the Wilson model

Table 2
Computational comparison of flowshop models

		CPU time (in seconds) for solving the problem by MILP model											
		Wagner			Wilson			Manne			Liao–You		
N[a]	M[a]	Mean	Std dev	Median	Mean	Std dev	Median	Mean	Std dev	Median	Mean	Std dev	Median
6	5	0.24	0.15	0.22	0.40	0.13	0.39	7.07	3.05	6.15	6.16	1.98	7.16
	7	0.58	0.17	0.59	0.93	0.22	0.83	9.89	2.66	10.76	7.83	1.57	8.68
	9	1.40	0.61	1.48	1.52	0.55	1.75	16.00	6.62	14.23	15.04	3.88	13.84
	Overall	0.74	0.61	0.55	0.97	0.59	0.77	10.99	5.66	10.76	9.68	4.69	8.68
7	5	0.63	0.28	0.66	1.30	0.48	1.37	39.85	34.86	22.47	37.19	28.14	28.28
	7	1.59	0.95	1.31	2.10	0.71	2.36	63.31	13.09	60.26	62.81	11.86	62.81
	9	3.40	2.63	2.80	4.78	2.59	4.01	116.86	10.09	124.41	85.66	26.57	75.96
	Overall	1.87	1.92	1.31	2.72	2.12	2.36	73.34	44.37	60.26	61.86	29.80	62.81
8	5	1.54	0.63	1.76	1.83	1.00	1.98	502.92	240.62	459.12	333.29	117.00	339.49
	7	3.61	1.95	4.67	5.54	3.90	5.93	643.67	338.09	598.63	487.13	171.34	508.22
	9	11.53	11.99	5.65	13.62	9.21	10.49	1163.86	527.80	1437.09	839.86	375.48	743.41
	Overall	5.56	7.88	3.07	7.00	7.40	5.87	770.15	464.08	608.08	553.41	317.40	471.98
9	5	3.02	0.95	3.30	5.66	3.06	5.61	2835.94	686.90	3249.83	2703.51	450.54	2699.15
	7	23.88	20.23	16.26	21.31	5.15	21.09	6255.46	2045.53	5899.05	5461.92	1644.52	5661.99
	9	27.35	10.27	30.32	33.68	8.06	35.81	5491.22	1712.27	6107.50	5076.86	1844.66	5146.52
	Overall	18.04	16.46	14.83	20.22	13.03	21.09	4860.89	2113.98	4524.65	4414.10	1843.08	3962.05
Grand overall		6.56	11.32	1.89	7.73	10.59	2.53	1428.84	2279.29	167.88	1259.76	2061.25	142.92

[a] N = number of jobs, M = number of machines; five problems per cell.

Table 3
Comparative statistics for the flowshop models

N[a]	M[a]	Wil[b]: Wag[c]	Man: Wag	LY: Wag	Man: Wil	LY: Wil	Man: LY	Wag < Wil[d]	LY < Man[e]
6	5	0.98	34.06	28.18	17.27	14.60	0.16	5	3
	7	0.63	16.33	13.32	9.84	7.93	0.29	5	4
	9	0.21	11.32	11.61	11.01	9.84	0.08	3	4
	Mean	0.61	20.57	17.70	12.70	10.79	0.17		
7	5	1.14	94.60	82.92	42.20	37.19	0.10	5	4
	7	0.52	50.15	55.85	31.12	33.94	0.05	4	3
	9	0.57	42.06	33.15	26.15	19.31	0.52	5	4
	Mean	0.75	62.27	57.31	33.16	30.15	0.22		
8	5	0.18	433.56	281.40	550.89	340.16	0.59	3	4
	7	0.48	208.42	197.72	166.19	153.23	0.34	5	4
	9	0.62	192.70	121.47	116.46	72.74	0.41	4	5
	Mean	0.43	278.22	200.20	277.85	188.71	0.44		
9	5	0.86	1008.755	998.08	603.84	611.36	0.05	5	2
	7	0.27	331.22	287.36	311.89	268.89	0.15	4	4
	9	0.34	227.11	218.99	177.04	166.98	0.13	4	3
	Mean	0.49	522.40	501.48	364.26	349.08	0.11		
Grand mean		0.57	220.87	194.17	171.99	144.68	0.24	52	44

[a] N = number of jobs, M = number of machines.
[b] Wag = Wagner; Wil = Wilson; Man = Manne; LY = Liao–You.
[c] Wil: Wag = (Wilson − Wagner)/Wagner; etc.
[d] Number of times in five problems Wagner model used less time than Wilson model.
[e] Number of times in five problems Liao–You model used less time than Manne model.

($p < 0.006$) for all cases. Likewise, the Wagner model uses less CPU time than the Wilson model for all cases when the cells are combined by values of N (new cell size = 15, $p < 0.018$).

Dichotomous constraints based models: *Manne versus Liao–You.* The values in the last column of Table 3 also indicate that the Liao–You model found the optimal solution in less time than the Manne model in 44 of the 60 problems solved for this paper. Since the other 16 problems were scattered among 11 cells, only in one of the 12 cells of the experiment ($M=9, N=8$, $p=0.03125$) was this difference statistically significant. When the cells are grouped by M, the time differences in the $M=7$ ($p=0.021$) and $M=9$ ($p=0.006$) cells are statistically significant. When grouped by N, the cells of $N=6$ ($p=0.059$), 7 ($p=0.059$), and 8 ($p=0.004$) have significant differences in computer solution times.

Overall comparisons. The average proportion deviations of the inferior models from the superior ones in each category with respect to other models are given in Table 3. For example, the notation Wil:Wag in Table 3 represents the statistic [time(Wilson)-time(Wagner)]/time(Wagner). Thus, for the cell ($M=5, N=6$), the Wilson model required, on average, $(0.98) \times 100$ or 98% more computer solution time than did the Wagner model, for the same five problems. Overall the Wilson model requires 57% more computer time, on average, than does the Wagner model for the problem sizes solved in this experiment. Likewise, the Manne model required about 24% more time, on average, than did the Liao–You adaptation proposed by Pan for the regular permutation flowshop model. Considering the low p values found in our statistical analysis of the experimental results and the fact that the CPU time required to solve each problem by various MILP models behaved consistently, the number of problem instances used in this paper is sufficient to rank various MILP models on the basis of CPU times.

For the regular permutation flowshop problem, the above discussion and the results summarized in Tables 2 and 3 clearly show the model superiority of the assignment problem based models relative to the dichotomous constraints based models. Thus, from a computational effort viewpoint, Pan's assertions stated earlier in this paper are incorrect.

We offer three possible explanatory factors for why, despite requiring more than twice the integer variables, the assignment problem based models (Wagner and Wilson) perform so much better than do the dichotomous constraints based models (Manne and Liao–You). First, the assignment problem based models, and particularly the Wagner model, consist of tightly linked sets of constraints, while the dichotomous constraints based models consist mostly of inequality constraints. Thus the branching on one binary variable in an assignment problem based model immediately affect the values of at least $2(N-1)$ other binary variables as well as a host of continuous variables. Second, the rank of the dichotomous constraints based models is much larger than that of the assignment problem based models, and matrix rank significantly affects the computational complexity of linear and integer programming problems. Third, the overall size of the matrices for the dichotomous constraints based models is much larger than that of the assignment problem based models; and this factor also can affect the computer solution times for the models.

3.2.2. *Anomalies in the means data*

There are three apparent anomalies in the summary data in Table 2 regarding mean solution times; and all three of these anomalies may be traced to just four of the 60 problems solved for this paper. First, for the Manne model, the mean solution time (5491.22 s) for the $M \times N = 9 \times 9$ cell is significantly smaller than the mean time (6255.46 s) for the 7×9 cell; and this difference goes against the logic of increased computer solution times as both M and N increase in size. The Manne model required significantly more computer time for three of the five problems in the 7×9 cell than it did for the other two problems. (The average was about 7550 s for these problems [2] 790, 791, and 794 versus an average of about 4315 s for problems 792 and 793.) Further, in the 9×9 cell, the Manne model required much less computer time for problem 990 (2871.43 s) than it did for the other four problems (average approximately 6146 s) in that cell. The combined result of the three *difficult* problems in the 7×9 cell with the *easy* problem in the 9×9 cell caused this apparent inversion of the cell means.

Second, a similar situation occurs in the $M \times N = 7 \times 9$ and 9×9 cells for the Liao–You model. This inversion of mean solution times with increasing M is again directly attributable to the Liao–You model having great difficulties with problems 790, 791 and 794 (mean = 6544 s) compared to problems 791 and 792 (mean = 3838 s). Further, the Liao–You model required a much shorter time (2581 s) to solve problem 990 than it did the others (mean = 5701 s) in that cell. The combined results again lead to an inversion of mean computation times with increasing number of machines.

Third, the Wagner model had a mean solution time less than that for the Wilson model for all experimental cells except $M \times N = 7 \times 9$, where the Wilson required 2.57 s less, on average, than did the Wagner model. This difference is directly attributable, again, to problem 790. The Wagner model required nearly a minute of CPU time to solve this problem while requiring just 14.89 s, on average, for the other four problems in this cell. On the other hand, the Wilson model required just 16.09 s to solve problem 790, and an average of 22.61 s to solve the other four problems. That is, the Wilson model had an easy time solving the most difficult problem in this cell, and a more difficult time solving the same four problems that the Wagner model solved easily.

This phenomenon of finding *difficult* problems in sets of randomly generated flowshop problems is not new to this paper. Ignall and Schrage [19] used a *most difficult* $M \times N = 3 \times 10$ problem to demonstrate their branch-and-bound approach to flowshop scheduling nearly 40 years ago. And similar problems appear in the MILP modeling research of Stafford [7], Tseng and Stafford [11], and Stafford and

[2] Each problem was coded as a three digit number where the first digit represents the number of machines, the second digit denotes the number of jobs and the last digit shows the problem number from 0 to 4.

Tseng [12,16]; and in research involving flowshop heuristics reported by Aggarwal and Stafford [20].

3.2.3. *Analysis of the medians*

Because of the anomalies with mean solution times discussed above, the median solution times for each model, for all 12 cells of the experiment, are also reported in Table 2. The medians are consistent throughout all 12 cells for the Wagner, Wilson, and Manne models. That is, the medians increase with increasing values of both N and M for these three models. Further, for the $M \times N = 7 \times 9$ cell, the median solution time for the Wagner model (16.26 s) is less than that for the Wilson model (21.09 s). Thus the behavior of median solution times between the Wagner and Wilson models—Wagner being less than Wilson—is consistent with the behavior of these two models with regard to mean computation times for all cells except the anomaly described above. At the same time, the $M \times N = 7 \times 9$ and 9×9 cells for the Liao–You model again show an *inversion* for the median solution times. In addition, the median times for the Liao–You model for cells 5×6, 5×7, and 7×7 are also greater than their counterparts for the Manne model. This is not consistent with the behavior of these two models with regard to mean solution times for these cells. (The Liao–You model had a smaller mean computation time than the Manne model for all 12 cells of the experiment.) Such inconsistencies are not unusual when comparing two groups of data with nearly identical mean values.

4. Conclusions

This paper provides, for the first time on a common set of problems, an empirical analysis of the four competing MILP models designed to solve the regular permutation flowshop problem to minimize makespan. This empirical evidence consists of computer solution times for an array of 60 problems. Each problem was solved four times, once for each model, on the same computer and using the same MILP software. The results of this investigation, presented in Tables 2 and 3, can be summarized as follows:

- For the regular permutation flowshop problem, both assignment problem based models—Wagner and Wilson—required less computer solution (CPU) time for solving each problem than do both dichotomous constraints based models—Manne and Liao–You. Further, the difference in CPU times increases dramatically with increases in both numbers of jobs and numbers of machines associated with a problem.
- For the assignment problem approach, the Wilson model used, on average, 57% more computer time than the Wagner model to solve the same problems.
- For the dichotomous constraints approach, the Manne model used, on average, 24% more computer time than the Liao–You model to solve the same problems.

Thus, the use of computer solution time as a criterion for the superiority of a MILP model reveals the following ranking of the four MILP problems:

a. Wagner's model is the best MILP formulation.
b. Wilson's model is second best MILP formulation.
c. Manne's model trails behind Laio–You's formulation.

The above rankings are different from those reported by Pan [9] and suggest that the use of the binary number of variables may not be sufficient to assess the computational complexity of various MILP formulations. Therefore, additional research is necessary to develop and investigate appropriate measures to assess a MILP model's computational complexity. For the permutation flowshop problems, this may involve increasing the scope of the investigation reported in this paper to include more MILP formulations, larger sample sizes, and larger problem sizes used in the experiments. The number of binary variables used by Pan [9] and the CPU time used in this paper could be just two of the many possible dimensions that affect the complexity ranking of various MILP models.

Based on our experiments, we now describe some fruitful directions for future research. First, replication of this study using some other modern MILP software such as CPLEX and larger problem sizes would be beneficial to verify the findings of this paper. Second, consideration of objective functions other than makespan would be interesting. Third, extension of the mathematical programming approaches to solve flowshop problems with a secondary criterion is both interesting and useful. Fourth, replication of this study for other scheduling problems, like job-shop and regular flowshop without utilizing the permutation schedules would be useful to assess the complexity of various MILP formulations. Finally, future studies that combine the mathematical programming approaches with the use of heuristics to provide upper bounds to the solution values and the use of solution lower bounds found through some other methods will be interesting and useful.

Acknowledgements

Constructive comments from three anonymous reviewers and the Editor-in-Chief, Professor Lawrence M. Seiford significantly improved the results and the presentation of the paper.

References

[1] Wagner HM. An integer linear-programming model for machine scheduling. Naval Research Logistics Quarterly 1959;6:131–40.

[2] Bowman EH. The schedule-sequence problem. Operations Research 1959;7:621–4.

[3] Manne AS. On the job-shop scheduling problem. Operations Research 1960;8:219–23.

[4] Gupta JND. The generalized n-job, m-machine scheduling problem. Opsearch 1971;8:173–85.
[5] Morten TE, Pentico DW. Heuristic scheduling systems. New York: Wiley; 1993.
[6] Baker KR. Introduction to sequencing and scheduling. New York: Wiley; 1974.
[7] Stafford EF. On the development of a mixed-integer linear programming model for the flowshop sequencing problem. Journal of the Operational Research Society 1988;39: 1163–74.
[8] Wilson JM. Alternative formulations of a flow-shop scheduling problem. Journal of the Operational Research Society 1989;40:395–9.
[9] Pan CH. A study of integer programming formulations for scheduling problems. International Journal of Systems Science 1997;28:33–41.
[10] Liao CL, You CT. An improved formulation for the job-shop scheduling problem. Journal of the Operational Research Society 1992;43:1047–54.
[11] Tseng FT, Stafford EF. Two MILP models for the N $\times$ M SDST flowshop sequencing problem. International Journal of Production Research 2001;39:1777–809.
[12] Stafford EF, Tseng FT. Two models for a family of flowshop sequencing problems. European Journal of Operational Research 2002;142:282–93.
[13] Johnson SM. Optimal two- and three-stage production schedules with set-up times included. Naval Research Logistics Quarterly 1954;1:61–8.
[14] Gupta JND. A review of flowshop scheduling research. In: Ritzman LJ, Krawjewski LJ, Berry WL, Goodman SM, Hardy ST, Vitt LO, editors. Disaggregation problems in manufacturing and service organizations. Hingham, MA: Martinus Nijhoff Publishers; 1979. p. 363–88.
[15] Stafford EF, Tseng FT. On "redundant' constraints in Stafford's MILP model for the flowshop problem. Journal of the Operational Research Society 2003;54:1102–5.
[16] Stafford EF, Tseng FT. On the Srikar–Ghosh MILP model for the N $\times$ M SDST flowshop problem. International Journal of Production Research 1990;28:1817–30.
[17] Daniel WW. Applied nonparametric statistics. Boston: Houghton-Mifflin Company; 1978.
[18] Rios-Mercado RZ, Bard JF. Computational experience with a branch-and-cut algorithm for flowshop scheduling with setups. Computers and Operations Research 1998;25:351–66.
[19] Ignall E, Schrage L. Application of the branch-and-bound technique to some flow-shop scheduling problems. Operations Research 1965;13:400–12.
[20] Aggarwal SC, Stafford EF. A heuristic algorithm for the flowshop problem with a common job sequence on all machines. Decision Sciences 1975;6:237–51.