J. Opl Res. Soc. Vol. 39, No. 12, pp. 1163–1174, 1988
Printed in Great Britain.
0160-5682/88 $3.00 + 0.00

On the Development of a Mixed-Integer Linear Programming Model for the Flowshop Sequencing Problem

EDWARD F. STAFFORD
University of Alabama in Huntsville, USA

This paper describes the development of a mixed-integer linear programming (MILP) model for the standard N-job, M-machine flowshop sequencing problem. Based on an earlier all-integer model developed by Wagner, this MILP model has been used to solve optimally problems with as many as 25 jobs and as many as 10 machines. Variants of the standard flowshop model, including a variety of performance measures, are also presented. Computational experience involving the successful solution of over 175 flowshop problems is discussed, and suggestions for future research projects are offered.

Key words: integer, linear, production, programming, scheduling, sequencing

INTRODUCTION

Since the early 1950s, many researchers have examined the standard flowshop sequencing problem and its many variations. With the current growing interest in group technology (GT), just-in-time manufacturing (JIT), computer-integrated manufacturing (CIM) and the like, manufacturing systems design is moving away from the jobshop orientation, and is moving towards a flowshop or hybrid jobshop orientation. Thus the importance of flowshop-oriented research promises to increase in the future.

Despite nearly 30 years of effort, flowshop-oriented researchers have had but little success in developing efficient algorithms for determining optimal solutions to the standard flowshop problem. Graves,[1] Gonzalez and Sahni[2] and many others attribute this lack of success to the fact that the flowshop problem is NP-complete. In addition, until recently, there has also been a lack of both computer hardware and software capable of supporting the development and testing of new modelling procedures for this rather complex systems scheduling problem.

This paper describes the development of a mixed-integer linear programming (MILP) model for the standard N-job, M-machine flowshop sequencing problem. Based on an earlier all-integer model developed by Wagner,[3] this MILP model has been used to solve *optimally* problems with as many as 25 jobs and as many as 10 machines. The MILP model changed the Wagner model in three simple, yet effective ways: (1) integer job- and machine-idleness variables were converted to real 'integer' variables; (2) a single variable and corresponding constraint were added to capture the value of the performance measure makespan directly; and (3) two sets of simple constraints were added to 'anchor' the sequence Gantt chart to the zero time line. Variants of the standard flowshop model, including a variety of performance measures and a MILP model for the flowshop where no intermediate queues of jobs are allowed, are also described. Computational experience involving the successful solution of over 175 flowshop problems is presented. Suggestions for future research projects are presented at the end of this paper. The description is presented as an historical narrative in order to document original authorship of various pieces of the model, and to establish dates of development of each of these pieces.

THE STANDARD FLOWSHOP PROBLEM

As a special case of the general jobshop problem, the standard flowshop problem has been of interest to researchers for more than 30 years. A brief review of this problem, and of the historical development of solution techniques for it, will facilitate an understanding of the MILP model described later in this paper.

Description of the problem

There are two major elements of the flowshop problem: (1) a production system of M machines; and (2) a set of N items or jobs to be processed on those machines. In the flowshop problem, all jobs are so similar in technological nature that they all have essentially the same order of processing on the set of machines. Thus the production system can be viewed as a 'pipeline', and the focus is on the problem of ordering or sequencing the jobs through the production system in order to minimize some measure of production cost.

The assumptions of the standard flowshop have been widely discussed in the production research literature,[4–7] and thus are only summarized here. All N jobs are available at time zero, at the beginning of processing of the entire jobset. Jobs will be processed as soon as possible on a given machine, and no passing of jobs is allowed. A single job may be assigned to at most one machine at a time, and at most, only one job can be processed at a time on a specific machine. The standard flowshop problem may then be summarized as:[7]

> '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.'

Approaches to problem solution

Solution approaches to the flowshop problem found in the literature may be classified as either optimization techniques or heuristic techniques. (A more comprehensive literature review is included in an expanded version of this paper,[8] available from the author.) Optimization techniques included complete enumeration,[4,9–11] simple algorithms,[12] mathematical programming,[3,13] branch-and-bound[14–17] and combinatorial search.[18–22] Heuristic sequencing techniques included those that develop: (1) a single final sequence;[23,24] (2) a single sequence which is then subjected to an 'improvement' technique to increase the likelihood of a near-optimal sequence;[4,25] and (3) several sequences, from which the best is chosen.[26] Studies by Dannenbring[25] and Stafford and Smith[27,28] indicated that single-sequence heuristics generally produced inferior sequences compared to other types of heuristics. Usually, the performance criterion for both these heuristics and for the optimization techniques has been jobset makespan.

THE WAGNER MODEL

In 1959, Wagner[3] described a variety of all-integer linear programming models for machine scheduling, including the three-machine permutation flowshop. It is appropriate to examine Wagner's flowshop model because the model described later in this paper is based on, and offers improvements to Wagner's original model. The notation used to present Wagner's model is based on the notation found in the Conway, Maxwell and Miller (CMM) text,[5] but it is slightly modified to eliminate both subscripts and superscripts on the same variable.

Presentation of the model

Wagner's model was limited to sequencing a set of N jobs on a flowshop of three machines, and this problem is often referred to as the $N/3/F/F_{max}$ problem. All variables were integer-valued. The Wagner model is summarized in Table 1.

Variables. Wagner defined Z as an $N \times N$ matrix of binary variables to represent the assignment of the set of N jobs to the N positions of the production sequence. The jth column of Z, a column vector to be used in the model constraints, is $Z(j) = [Z_{1j}, Z_{2j}, \ldots, Z_{Nj}]$. The machine idle-time variables (X_{jk}) and job idle-time variables (Y_{jk}) were to be assigned integer values.

Processing-time coefficients. The row vectors A, B and D were used to represent the processing times of the set of N jobs on machines 1, 2 and 3, respectively. For example, $A = (a_1, \ldots, a_N)$ = processing times for jobs $1, \ldots, N$ on machine 1. Vectors B and D were similarly defined.

Constraints. Wagner presented four sets of constraints to represent the essence of the flowshop problem. The first two sets [equations (2) and (3) in Table 1], identical to the constraints of the linear programming model of the assignment problem, ensured that: (1) each job is assigned to one, and only one position in the production sequence; and (2) each position in the sequence is occupied by one and only one job.

TABLE 1. *Summary of the Wagner integer linear programming model for the standard flowshop sequencing problem*

Decision variables

Z_{ij} — 1, if ith job is assigned to jth position in sequence; 0, otherwise. $(i, j = 1, \ldots, N)$

X_{jk} — Idle time on machine k before the start of the job in position j in the sequence of jobs. $(j = 1, \ldots, N;\ k = 1, 2, 3)$

Y_{jk} — Idle time of job in jth position in sequence, after finishing processing on machine k, while waiting for machine $k + 1$ to become free. $(j = 1, \ldots, N;\ k = 2, 3)$

Objective function

$$\text{Minimize: } \sum_{j=1}^{N} X_{j,3} \tag{1}$$

Constraints

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

$$\sum_{j=1}^{N} Z_{ij} = 1 \quad (j = 1, 2, \ldots, N) \tag{3}$$

$$A * Z(j+1) + Y_{j+1,1} - Y_{j,1} - B * Z(j) - X_{j+1,2} = 0 \quad (j = 1, \ldots, N-1) \tag{4}$$

$$X_{j+1,2} + B * Z(j+1) + Y_{j+1,2} - Y_{j,2} - D * Z(j) - X_{j+1,3} = 0 \quad (j = 1, \ldots, N-1) \tag{5}$$

Processing-time coefficients

A, B, D — Vector $(1 \times N)$ of job processing-times on machines 1, 2 and 3 respectively.

The second two sets of constraints provided the Gantt-chart representation of equal time-slices on adjacent machines, for all job positions in the sequence. This 'Gantt-chart accounting' for corresponding time-slices on machines 1 and 2 was represented as shown in equation (4) in Table 1, and for corresponding time-slices on machines 2 and 3, in equation (5) in Table 1.

Objective function. Wagner argued that minimizing the idle time on machine 3 of the production system was equivalent to minimizing makespan or maximum flowtime, $F_{\max}$. Thus his objective function was to minimize the sum of the $X_{j,3}$ variables as shown in equation (1) in Table 1.

Discussion of the model

In subsequent studies with Story[29] and with Giglio,[30] Wagner described computational results and difficulties with his model. They reported that they were only able to solve problems with $N \leqslant 6$, and that some of these small problems resisted solution for more than 24 hr of computer time. These researchers were severely restricted by the computer tools available at the time: (1) relatively slow hardware; and (2) integer programming software of the cutting plane variety, which has proved significantly inferior to more recent branch-and-bound integer programming codes such as MPSX[31] and LINDO.[32] This model was abandoned in the early 1960s for other optimization and heuristic techniques.

More important to this paper, the Wagner model had at least four other weaknesses. First, it was limited to a production line of three machines. Second, all variables in Wagner's model were integer variables. Third, Wagner's model was incomplete structurally, and this allowed incorrect solutions to certain problems to be called 'optimal' solutions by commercial integer programming codes. Fourth, a user of Wagner's model would have to reconstruct the complete Gantt chart of the sequence indicated by the optimal solution in order to determine the optimal value of makespan for the solved problem. All of these weaknesses are eliminated by the MILP flowshop model described below.

DEVELOPMENT OF THE MILP FLOWSHOP PROBLEM MODEL

Development of the MILP flowshop model as an alternative to Wagner's all-integer linear programming model for the $N \times 3$ flowshop began in 1972, and experimentation with it has continued to the present. This narrative is presented in three parts, each corresponding to a major step in the development or testing of the model.

Part I: Testing Wagner's model, 1972

Starting in March 1972, this author began a research project to solve flowshop problems using the Wagner integer programming model. The IBM software BBMIP,[33] run on an IBM 360/67 computer, would not yield even a first feasible integer solution beyond the LP starting solution for a 3×3 problem.

Introduction of real variables. The failure of BBMIP to produce integer solutions led to a search for alternative software. A zero-one mixed-integer linear programming program, code-named the 'Brazil Code', was made available by an anonymous colleague. In order to use this Brazil software, the Wagner model was modified by converting the integer X- and Y-variables to real-valued variables. Further, the job processing times on all machines were set to 'integer' values. (The linear programming approach to solving the transportation problem uses the same techniques to ensure 'integer' values for the real-valued decision variables in the optimal solution.)

An integer feasible starting solution was thus obtained by the LP subroutines of the Brazil code, but the entire solution became infeasible when the branch-and-bound subroutines attempted to optimize this starting solution. Although this investigation was temporarily abandoned at the time, the conversion of the integer X- and Y-variables to real variables was retained for further investigations.

Part II: Developing the model, 1978

In 1977, after a career move, this author returned to the investigation of the mixed-integer version of the Wagner model. The MILP model was *generalized to* an *N-job*, *M-machine* model during the spring of 1978. Additional elements that corrected the difficulties found with the Wagner formulation were not identified until February 1980, when MPSX became available to the author. For convenience, the MILP model is summarized in Table 2.

Decision variables. The definitions of the decision variables used in the MILP model are basically the same as those of the original Wagner model, but the X_{jk} and Y_{jk} variables were converted to real linear variables. Further, for the X-variables, $k = 1, 2, \ldots, M$ (M machines in the production system); and for the Y variables, $k = 1, 2, \ldots, M-1$. A single additional variable was added to this expanded set of decision variables:

X_F = makespan or maximum flowtime of any job in the jobset; all jobs processed on the M machines in the same technological order.

The single variable eliminated the need to reconstruct the entire Gantt chart of a Wagner solution in order to determine the value of the optimal makespan for a set of N jobs processed on M machines.

Processing-time coefficients. A revised notation for the processing-time coefficients was used to generalize Wagner's model to M machines. Let t_{RS} be the processing time of job S on machine R. Then $T = \{t_{RS}\}$ is an $M \times N$ matrix of processing times of the entire jobset of N jobs on all of the M machines in the production system. Additionally, $T_R = [t_{R,1}, \ldots, t_{R,N}]$ is the row vector of processing times of all jobs on machine R.

TABLE 2. *Summary of the mixed-integer linear programming model (MILP) for the standard flowshop sequencing problem*

Decision variables

Z_{ij}	1, if ith job is assigned to jth position in sequence; 0, otherwise. ($i, j = 1, \ldots, N$)
X_{jk}	Idle time on machine k before the start of the job in position j in the sequence of jobs. ($j = 1, \ldots, N$; $k = 1, \ldots, M$)
Y_{jk}	Idle time of job in jth position in sequence, after finishing processing on machine k, while waiting for machine $k+1$ to become free. ($j = 1, \ldots, N$; $k = 1, \ldots, M-1$)
X_F	Makespan or maximum flowtime of any job in the jobset; all jobs processed on the M machines in the same technological order.

Objective function

$$\text{Minimize makespan} = X_F \tag{8}$$

Constraints

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

$$\sum_{i=1}^{N} Z_{ij} = 1 \quad (j = 1, 2, \ldots, N) \tag{3}$$

$$T_R * Z(j+1) + Y_{j+1,R} + X_{j+1,R} - Y_{j,R} - T_{R+1} * Z(j) - X_{j+1,R+1} = 0 \quad (j = 1, \ldots, N-1)(R = 1, \ldots, M-1) \tag{6}$$

$$X_F - \sum_{j=1}^{N} [T_M * Z(j)] - \sum_{j=1}^{N} X_{j,M} = 0 \tag{7}$$

$$Y_{1,k} = 0, \quad (k = 1, \ldots, M-1) \tag{13}$$

$$X_{l,k} - \left\{\left[\sum_{p=1}^{k-1} T_p\right] * Z(1)\right\} = 0 \quad (k = 2, \ldots, M) \tag{14}$$

Constraints. The constraints sets for the MILP $N \times M$ generalization of Wagner's model are presented in Table 2. Equations (2) and (3), which provide assignment of jobs to sequence positions, are retained from the Wagner model. Equations (4) and (5) of Wagner's model are replaced by the following equation, which provides Gantt-chart accounting between all adjacent pairs of machines in the M-machine system:

$$T_R * Z(j+1) + Y_{j+1,R} + X_{j+1,R} - Y_{j,R} - T_{R+1} * Z(j) - X_{j+1,R+1} = 0$$
$$(j = 1, \ldots, N-1)(R = 1, \ldots, M-1). \quad (6)$$

The following equation was added to determine makespan of an optimal solution without having completely to reconstruct the Gantt chart for that solution:

$$X_F - \sum_{j=1}^{N} [T_M * Z(j)] - \sum_{j=1}^{N} X_{j,M} = 0. \quad (7)$$

The $X_{j,1}$ variables, which were set to zero in the Wagner model and thus were eliminated from equation (4) were also set to zero in equation (6) of the MILP model.

Objective function. With the addition of equation (7), the objective function of the MILP model for the $N \times M$ flowshop is

$$\text{minimize: } X_F. \quad (8)$$

This completed the generalization of Wagner's all-integer model of the $N \times 3$ flowshop to a mixed-integer model of the $N \times M$ flowshop.

Extensions of the MILP model. At the end of spring 1978, with no large-scale integer programming software available to test the MILP $N \times M$ flowshop model, attention turned to adapting this model to variants of the standard flowshop model that had been reported in the literature.

Alternative performance measures. Makespan had been the usual flowshop schedule performance measure since the seminal work of Johnson in the 1950s. At the same time, reasonable alternative performance measures had been proposed. Three of these, developed as extensions of the MILP model, are shown in Table 3.

I. Machine idle-time: Minimizing machine idle-time of a flowshop is equivalent to maximizing the efficiency of productive resources (machines, labour). Two variants of this performance measure were identified. Equation (9) represents machine idle-time overall, and equation (10) minimizes machine idle-time only within the sequence, ignoring the time that machines $2, \ldots, M$ wait for the arrival of the first job in the sequence.

II. Job idle-time: Managers of flowshops might also have concerns regarding the amount of time jobs spend waiting for the next machine in the sequence to become free. Equation (11) in Table 3 is the expression to minimize job idle-time.

III. Mean flowtime: Conway *et al.*[5] discussed the minimization of average or mean flowtime for the two-machine flowshop. Panwalkar and Khan[34] and Bansal[35] described solution approaches

TABLE 3. *Alternative performance measures for the MILP* N × M *standard flowshop sequencing problem*

Machine idle-time

MI1:* Minimize: $\sum_{k=2}^{M} \sum_{j=1}^{N} X_{jk}$ (9)

MI2:† Minimize: $\sum_{k=2}^{M} \sum_{j=2}^{N} X_{jk}$ (10)

Job idle-time

JI:‡ Minimize: $\sum_{k=1}^{M-1} \sum_{j=1}^{N} Y_{jk}$ (11)

Sum of flowtimes

Minimize: $S_f = \sum_{j=1}^{N} \{(N-j+1)(X_{j,M})\} + \sum_{j=1}^{N} \{(N-j+1)[T_M * Z(j)]\}$ (12)

*Includes time machines wait for first job in sequence.
†Includes machine idle-time within sequence only.
‡Includes job idle-time only for jobs that have started processing.

for the M-machine flowshop using this performance measure, and Szwarc[36] developed several other properties of this problem. Flowtime, or job-completion time, is the total time that a job spends in a production facility. Let F_j represent the flowtime of the job that is assigned to position j in the sequence. Then it can be shown that minimizing mean flowtime, F_{BAR}, is equivalent to minimizing the sum of job flowtimes, S_f, also called sum of job-completion times. The derivation for the expression for S_f [equation (12), Table 3] is presented elsewhere.[8] Any of these alternative performance measures, equations (9)–(12), may be substituted for equation (8) in the MILP formulation of the $N \times M$ flowshop model.

Part III: Correcting the Wagner model, 1979–1980

By December 1978, all details of the MILP alternative to Wagner's all-integer linear programming model of the flowshop, described to this point, had been developed. In addition, a MILP model for the special case of the flowshop wherein no intermediate job queues are allowed[37–42] was formulated during the summer of 1978. Experiments conducted with this NIQ flowshop model are described elsewhere.[43]

In late 1979, powerful software with which to test these models became available when the IBM integer programming package MPSX[31] was leased for faculty use at the university where this author was then employed. On 25 February 1980, MPSX was used in an attempt to solve the 3×3 flowshop problem shown in Table 4, using the MILP version of Wagner's model. The MPSX solution did not match either of the optimal solutions shown in Table 4. In addition, $Y_{1,1}$, which should have been zero, had a value of 1. This caused other X- and Y-variables to have incorrect values. After carefully examining the Gantt charts of the MPSX solution, it was determined that the Wagner model needed to be 'anchored' to the Y-axis of the Gantt chart (time = 0).

At the time that the MILP $N \times M$ model was being developed, it was noted that Wagner's model included an $X_{1,3}$ variable in the objective function, yet that variable did not appear in any constraint of his model. Thus $X_{1,3}$ would be assigned a value of zero in any solution, even though machine 3 *would* be idle waiting for the arrival of the first job in the sequence. This was an additional indication that Wagner's model was incomplete structurally.

Two new sets of constraints were added to the MILP model described above. The first, which ensured that the first job in the sequence would always pass immediately to each successive machine in the production line in accordance with the assumptions of the model, may be represented as

$$Y_{1,k} = 0, \quad (k = 1, \dots, M-1). \tag{13}$$

The second set of constraints anchors the Gantt chart to the Y-axis by accounting for machine idle-time of the second and successive machines while they await the arrival of the first job in the sequence. This set of constraints may be represented by

$$X_{1,k} - \left\{ \left[\sum_{p=1}^{k-1} T_p \right] * Z(1) \right\} = 0 \quad (k = 2, \dots, M). \tag{14}$$

TABLE 4. *Sample 3 × 3 flowshop problem with enumerated solutions*

		Processing times Job 1	2	3
Machine	1	3	4	2
	2	1	5	7
	3	2	1	7

Sequence of jobs	Makespan	Mean flowtime	Job idle-time	Machine idle-times	
1-2-3	23	14.00	3*	12†	19‡
1-3-2	17	13.00	2	4	11
2-1-3	21	14.33	3	2	15
2-3-1	22	17.33	13	3	16
3-1-2	16	14.67	8	2	10
3-2-1	16	14.33	6	0	8

*Total job delay, once jobs have started on machine 1.
†Total machine idle-time within sequence.
‡Includes time waiting for first job in sequence.

The effect of equation (14) is to ensure that the time each machine waits for the first job in the sequence is equal to that job's total processing time on all previous machines in the production line.

The MILP flowshop model, augmented by equations (13) and (14), was used to solve the 3×3 problem from Table 4 on 27 February 1980, using MPSX on an IBM 370/168 computer. The results were identical to those shown in Table 4. The first integer solution was the sequence 1-3-2 ($X_F = 17$). The optimal sequences 3-1-2 and 3-2-1 ($X_F = 16$) were each detected in subsequent iterations. Careful analyses of the MPSX solutions indicated that all values of all variables agreed completely with those values determined by hand-charting each solution.

Additional problems were created and solved during the next 2 months, including the 'most difficult' ten-job, three-machine problem reported by Conway *et al.*[5] After successfully solving the CMM problem, two points were made quite clear: (1) the MILP flowshop model would work for problems larger than those successfully solved by Wagner and cohorts; and (2) if extensive experiments were to be conducted on the MILP model, then a front-end program to generate MPSX data sets would be needed. At that point, a career move temporarily interrupted the continuation of this research.

ANALYSIS AND EXPERIMENTATION WITH THE MODEL

The MILP flowshop model described above and the NIQ variant of that model have been used in a variety of experiments to solve over 200 different flowshop problems. Some of these experiments are presented in this section in order to demonstrate the ability of this model optimally to solve relatively large flowshop problems, and to provide estimates of computer time needed to solve these problems. First, the model is analysed with respect to problem size and to the computer resources required efficiently to solve intermediate-sized problems.

Problem-size calculations

The number of constraints (rows) and variables (columns) that a mathematical programming software package can accommodate is a typical means of rating such packages. The number of constraints, NC, and the number of variables, NV, needed to formulate an N-job, M-machine standard flowshop problem using the MILP model may be calculated as

$$\mathrm{NC} = N + M(N-1) + 1 \tag{15}$$

$$\mathrm{NV} = N(N + 2M - 1). \tag{16}$$

For example, a ten-job, seven-machine problem would require 74 constraints and 230 variables. An additional row would be required for the objective function.

Problem/data set generating program

In the autumn of 1980, a FORTRAN program was written to generate formatted MPSX data sets for standard flowshop problems with known processing times. In addition, this program would generate any number of problems with randomly distributed processing times for use in research projects. A similar program was written to generate problems for the NIQ flowshop model. These programs were used to generate the problems used in the experiments described below.

Experiment I: Solving the CMM 'most difficult' problem

In March 1980, the CMM[5] 'most difficult' problem was solved on an IBM 370/168 computer using MPSX and the MILP flowshop model. Computer run-time (CPU) to solve this ten-job, three-machine problem was 4.69 min. As claimed by CMM, the minimum makespan was 66 time units. When compared to computer solution-times for similar-sized problems, such as those described below, the solution time for this problem seemed excessive. A complete examination of the MPSX printout for this problem indicated 31 different, equally optimal sequences, each with a makespan of 66. Hence significant additional CPU time was needed to identify these many optimal solutions.

Experiment II: Evaluating flowshop heuristics

In January 1983, MPSX was once again available to this author for a period of approx. 4 months. One of the first extensive experiments conducted on the MILP model, using MPSX, had two purposes: (1) to determine CPU time requirements for problems with as many as 25 jobs; and (2) to evaluate the quality of solutions provided by heuristic flowshop sequencing techniques.

The experimental design for this project is presented in Table 5. Ten problems were generated and solved for each of the 12 experimental cells. Each problem solved was limited to a production line of three machines in order that the results would be directly comparable to the results of the Wagner projects and to other results reported in the literature. The factor 'range of the randomly distributed processing times' for jobs on machines was investigated in response to an anonymous referee of an earlier paper by this author, who claimed that this range biased results in favour of certain heuristics. Processing times were randomly generated from the uniform distribution. All 120 problems were solved on an IBM 3033 computer. The average CPU time required to solve the problems of each cell, in *seconds*, is shown in Table 5. Correlation analysis indicated no relationship between the range of processing-time values and the computer resources needed to solve the problem.

These same 120 problems were also solved with the CDS heuristic[28] and the RAES heuristic of Dannenbring.[25] The complete analyses of the solutions provided by the CDS and RAES heuristics are documented elsewhere.[44] Suffice it to say that the RAES heuristic found the optimal solution to 112 problems (93.3%), and the CDS heuristic, 76 problems (63.3%).

Experiment III: Solving the N × M *problem*

In January 1983, an experiment was conducted to investigate the computational resources needed to solve the N-job, M-machine standard flowshop. Three levels of jobs (5, 8, 10) and three levels of machines (3, 5, 7) were selected for this two-factor experiment. Three problems were solved for each of the nine cells. The job-processing times were uniformly distributed, ranging from one to ten time-units. All problems were solved on an IBM 3033 computer using MPSX. The CPU solution-time, makespan and number of different optimal solutions found are presented in Table 6 for all problems. A parallel study of the $N \times M$ MILP NIQ flowshop model is described elsewhere.[43]

Experiment IV: 'Project demo'

The MPSX mathematical programming software package allows the user to include several 'functions' in the formulation of any linear or integer programming problem. One of these functions would serve as the objective functions to be optimized. The other functions could be used to compute various measures of interest, such as departmental costs, labour and machine costs for certain products, and the like. In addition, the user can solve the same problem several times in a single computer run, switching the objective function for each new solution. This feature was thought to have promise for flowshop sequencing research using the MILP model, and the following experiment was designed and executed in February 1983 to demonstrate the multi-solution capabilities of MPSX.

TABLE 5. *Computer CPU times for solving* N × 3 *flowshop problems with the MILP model using MPSX*

Number of jobs	Range of processing times 1–10	1–50	1–100
5	2.06*	2.10	2.19
	(0.54)†	(0.51)	(0.40)
10	6.48	4.68	7.04
	(5.61)	(1.68)	(5.23)
15	11.55	39.03	23.29
	(8.92)	(65.50)	(21.52)
25	79.21	119.55	81.44
	(37.57)	(79.63)	(55.90)

*Mean CPU time of 10 jobs per experimental cell, in seconds.
†Standard deviation of CPU time.

TABLE 6. *Computer solution results from the initial MILP* N × M *standard flowshop problem experiment*

		Machines					
		3		5		7	
Jobs	Problem	CPU	F_{max}	CPU	F_{max}	CPU	F_{max}
5	1	5.38*	48† (1)‡	8.91	66 (8)	6.45	63 (1)
	2	6.18	40 (2)	5.58	56 (1)	7.42	62 (2)
	3	7.53	45 (6)	7.60	58 (4)	7.66	70 (3)
	Average	6.36		7.36		7.18	
8	1	1.84	55 (1)	30.73	71 (8)	160.55	84 (19)
	2	1.93	47 (1)	23.52	63 (3)	22.53	79 (5)
	3	2.19	61 (1)	19.92	67 (2)	10.90	82 (1)
	Average	1.99		22.39		64.66	
10	1	236.96	57 (2)	732.52	84 (99)	283.91	98 (3)
	2	7.56	67 (1)	731.82	80 (6)	191.43	92 (7)
	3	10.90	56 (1)	42.77	72 (47)	476.05	101 (1)
	Average	85.14		502.70		313.13	

*Computer CPU solution time, in seconds, on an IBM 3033.
†Optimal makespan value.
‡Number of different optimal solutions found.

Three problems of different sizes were randomly generated with the FORTRAN programs described above, and each problem was solved four different times according to the following 'models': (I) standard MILP flowshop problem, with minimum makespan as the objective; (II) standard MILP flowshop problem, optimized four times in a single run, once for each performance measure—makespan, two measures of machine idleness, and job idle-time; (III) NIQ MILP flowshop problem, optimized three times in a single run, once for makespan and once for each measure of machine idleness; and (IV) NIQ MILP flowshop problem, with minimum makespan only as the optimized performance measure.

The results of this 'project demo' are presented in Table 7. For each problem, the computer CPU time shown for model I represents the amount of time required to get the first optimal solution completed in model II. The difference in CPU times, model II − model I, represents the time required to obtain the additional three optimizations for model II. The same relationship holds

TABLE 7. *Computer results from solving multiple performance measures for 'project demo' problems*

Problem	model	CPU time	Performance measure	Optimal value	F_{max}†	MI_1‡	MI_2§	JI¶
5 × 3*	I	7.40	Makespan	33 (1)	33			
	II	9.21	Makespan†	33 (1)	33	14	5	28
			Machidl1‡	7 (2)	33	7	0	19
			Machidl2§	0 (1)	39	17	0	29
			Jobidle¶	1 (2)	38	18	11	1
	III	9.49	Makespan	35 (1)	35	27	18	
			Machidl1	20 (1)	40	20	11	
			Machidl2	11 (1)	40	20	11	
	IV	7.76	Makespan	35 (1)	35	27	18	
6 × 4	I	7.31	Makespan	51 (1)	51			
	II	14.24	Makespan	51 (1)	51	39	16	17
			Machidl1	33 (1)	51	33	10	26
			Machidl2	8 (1)	60	57	58	8
			Jobidle	8 (1)		44	21	8
	III	16.58	Makespan	54 (1)	54	57	34	
			Machidl1	53 (1)	56	53	30	
			Machidl2	30 (1)	56	53	30	
	IV	9.25	Makespan	54 (1)	54	57	34	
8 × 5	I	20.82	Makespan	67 (2)	67			
	II	83.59	Makespan	67 (2)	67	72	23	68
			Machidl1	63 (5)	67	63	14	53
			Machidl2	7 (3)	73	82	7	71
			Jobidle	32 (2)	74	88	52	32
	III	267.67	Makespan	78 (12)	78	142	87	
			Machidl1	118 (1)	81	118	82	
			Machidl2	82 (1)	81	118	82	
	IV	65.69	Makespan	78 (12)	78	142	87	

*N jobs × M machines.
†Equation (8), makespan of jobset.
‡Equation (9), overall machine idle-time.
§Equation (10), machine idle-time within sequence.
¶Equation (11), total job idle-time within sequence.

between the CPU times shown for models III and IV. This ability simultaneously to measure multiple performance measures, and sequentially to optimize different performance measures for the same problem would prove useful in a multi-criteria flowshop problem model.

FUTURE DIRECTIONS

Since the completion of the work reported in this paper, only a few new studies have surfaced in the literature that relate to the MILP model. This section of the paper reviews two of these studies, and it provides some potential directions for future research projects.

Related studies

Two different studies from the recent literature were reviewed for relevancy to this paper.

Another MILP model? In a recent meeting, Booth and Turner[45] reported analysing several heuristics for the standard flowshop sequencing problem. In addition, they suggested that the mathematical programming model reported by French[46] was a viable alternative to the MILP model reported in this paper. French's model was essentially the original Wagner model, but with somewhat modified notation. His formulation of the 'Gantt-chart accounting' constraints would allow an extension of Wagner's model to M machines, but at the same time, it retained the weaknesses of the Wagner model that the MILP model eliminated. French did not report any problem solution studies using his flowshop formulation.

A goal programming extension. In the spring semester 1981, W. J. Selen, then a graduate student in this author's doctoral seminar on production planning and control, was assigned the task of solving several flowshop sequencing problems using the MILP model described above, for both the standard and the NIQ flowshops. In addition, partially to fulfil the requirements of another graduate seminar, Selen was to append the MILP model with goal programming.

In his report on this project, Selen[47] described solving a single 3×3 flowshop problem using the MINT code,[33] but he could not solve this same problem with the integer goal programming code of Arthur and Ravindran.[48] (This problem was the same problem used by this author in 1980 to first prove the feasibility of the MILP model. As can be seen from Table 4, this problem can be quickly solved *by hand* for any performance measure and for any combination of measures under a multiple-criteria decision-making environment.)

Selen and Hott[49] recently reported solving a six-job, four-machine standard flowshop sequencing problem using LINDO[32] to solve the MILP model described above and the goal programming extension developed by Selen. The Selen and Hott[49,50] goal programming extensions to the MILP flowshop model involve the addition of several deviational variables and constraints. Further, the analyst must interact with the software (LINDO or MPSX) several times to arrive at a final 'optimal solution' wherein various competing goals have been satisfied in the proper weighted order. As shown in Table 7, the same results can be achieved without the addition of the deviational variables, and often with much less effort, if the alternative performance measures shown in Table 3 are incorporated as 'functions' (non-constraining relationships) within either LINDO or MPSX solution models for a flowshop problem. Thus goal programming is not the only alternative for seeking a multi-criteria solution to the MILP flowshop problem.

Testing other hardware and software

MPSX, the software used to solve nearly all of the problems described in this paper, is expensive to lease, and it is not always available to interested researchers. In addition, the user has only limited control over the solution procedure, and for large problems, MPSX provides excessive unneeded lines of printout. Software is needed that is transportable to other sizes and brands of computers.

High-powered mini- and main-frame computers are becoming very common across university campuses. A major study needs to be conducted wherein the same software would be used to solve the same set of flowshop problems on as wide a variety of these machines as possible. Comparison of computation times and solution success would help bench-mark each of these computers relative to the others. A problem generator similar to the FORTRAN code described above would be needed to eliminate the excessive burden of preparing problem data sets for any software chosen. Currently, a similar code is being developed to generate problems to be solved with LINDO.

The computational speeds claimed for supercomputers suggest that these computers can move the solution limits of flowshop sequencing problems well beyond the limits of most real problems found in industry, despite the problem being NP-complete. The next major step is to identify available software that is user-friendly and that will work on such powerful computers.

Microcomputers should also be examined. Recently, Stafford and Tyagi[51] successfully solved the 3×3 problem shown in Table 4 on an NCC/286 personal computer using a small version of LINDO. Solution time was less than 2 sec. The limits of this software/hardware combination will be tested after the LINDO problem generator is completed.

A major experiment is needed to test the relative capabilities of currently available integer programming software, and the MILP flopshop model is sufficiently complex to identify only those packages that should be considered by serious researchers and problem-solvers.

Improved solution times through bounding

At present, the MILP flowshop model provides no mechanism by which known upper bounds on the integer solutions can be introduced to speed solution of the problem. Lageweg[17] described a variety of bounding procedures for the flowshop problem. Several, including those proposed by Bansal,[35] were lower bounds, and thus would be of no use in speeding a branch-and-bound MILP solution. As shown by Stafford,[44] the CDS algorithm would yield a near-optimal makespan in very little computational time. A single extra constraint,

$$X_F \leqslant \text{CDS minimum makespan}, \tag{17}$$

would allow the upper bound on makespan to be easily added to the current MILP model. Research is needed to determine if such bounding would speed solution time significantly.

CONCLUSION

A mixed-integer linear programming model for the standard flowshop has been developed to overcome the difficulties encountered with the earlier all-integer Wagner model. In addition, a variant of this model has been developed to solve flowshop problems wherein jobs cannot be delayed once they are started into the manufacturing sequence. A variety of expressions for different performance measures was also developed. Over 200 problems have been formulated by these models and solved using commercially available software. Solution times were quite reasonable, and the feasibility of these models has been well tested. A variety of future research projects was outlined.

REFERENCES

1. S. G. Graves (1981) A review of production scheduling. *Mgmt Sci.* **29,** 646–675.
2. T. Gonzalez and S. Sahni (1978) Flowshop and jobshop schedules: complexity and approximation. *Opns Res.* **26,** 36–52.
3. H. M. Wagner (1959) An integer linear-programming model for machine scheduling. *Nav. Res. Logist. Q.* **6,** 131–140.
4. S. C. Aggarwal and E. F. Stafford (1975) A heuristic algorithm for the flowshop problem with a common job sequence on all machines. *Dec. Sci.* **6,** 237–251.
5. R. W. Conway, W. L. Maxwell and L. W. Miller (1967) *Theory of Scheduling*. Addison–Wesley, Reading, Mass.
6. K. R. Baker (1974) *Introduction to Sequencing and Scheduling*. Wiley, New York.
7. J. N. D. Gupta (1972) Heuristic algorithms for multistage flowshop scheduling problem. *AIIE Trans.* **4,** 11–18.
8. E. F. Stafford (1988) On the development of a mixed-integer linear programming model for the standard flowshop. Working Paper 88–02, College of Administrative Science, University of Alabama in Huntsville.
9. E. F. Stafford (1982) A machine-based lower bound estimate of makespan for flowshop scheduling problems. Paper presented at the *SE AIDS Meetings*, Memphis, Tenn.
10. E. F. Stafford (1983) Comparison of improvement routines for heuristic flowshop sequencing procedures. Paper presented at the *ORSA/TIMS National Meeting*, Orlando, Fla.
11. E. F. Stafford (1983) Lower bound analyses of flowshop scheduling procedures. Paper presented at the *ORSA/TIMS National Meeting*, Orlando, Fla.
12. S. M. Johnson (1954) Optimal two- and three-stage production schedules with set-up times included. *Nav. Res. Logist. Q.* **1,** 61–68.
13. E. H. Bowman (1959) The schedule-sequencing problem. *Opns Res.* **7,** 621–624.

14. Z. A. LOMNICKI (1965) A branch-and-bound algorithm for the exact solution of the three machine scheduling problem. *Opl Res. Q.* **16,** 89–100.
15. E. IGNALL and L. SCHRAGE (1965) Application of the branch-and-bound technique to some flow-shop scheduling problems. *Opns Res.* **13,** 400–412.
16. S. S. PANWALKAR and A. W. KHAN (1975) An improved branch and bound procedure for the $n \times m$ flow shop problems. *Nav. Res. Logist. Q.* **22,** 787–790.
17. B. J. LAGEWEG (1978) A general bounding scheme for the permutation flow-shop problem. *Opns Res.* **26,** 53–67.
18. R. A. DUDEK and O. F. TEUTON (1964) Development of M-stage decision rule for scheduling n jobs through M machines. *Opns Res.* **12,** 471–497.
19. K. R. BAKER (1975) An elimination method for the flow-shop problem. *Opns Res.* **23,** 159–162.
20. W. SZWARC (1971) Elimination methods in the $m \times n$ sequencing problem. *Nav. Res. Logist. Q.* **18,** 295–305.
21. W. SZWARC (1973) Optimal elimination methods in the $m \times n$ flow-shop scheduling problem. *Opns Res.* **21,** 1250–1259.
22. J. N. D. GUPTA (1975) Analysis of a combinatorial approach to flowshop scheduling problems. *Opl Res. Q.* **26,** 431–440.
23. E. S. PAGE (1961) An approach to scheduling of jobs on machines. *J. R. Statist. Soc.* (Series B) **23,** 484–492.
24. D. S. PALMER (1965) Sequencing jobs through a multi-stage process in the minimum total time—a quick method of obtaining a near optimum. *Opl Res. Q.* **16,** 101–107.
25. D. G. DANNENBRING (1977) An evaluation of flow shop sequencing heuristics. *Mgmt Sci.* **23,** 1174–1182.
26. H. G. CAMPBELL, R. A. DUDEK and M. L. SMITH (1970) A heuristic algorithm for the n job M machine sequencing problem. *Mgmt Sci.* **16,** B630–B637.
27. E. F. STAFFORD and A. E. SMITH (1977) Comparison of recent heuristic flowshop techniques across multiple criteria. Paper presented at the *ORSA/TIMS National Meetings*, Atlanta, Ga.
28. E. F. STAFFORD and A. E. SMITH (1979) Multivariate analyses of heuristic solutions to large-scale flowshop scheduling heuristics. Paper presented at the *TIMS XXIV International Meeting*, Honolulu, Hawaii.
29. A. E. STORY and H. M. WAGNER (1963) Computational experience with integer programming for job-shop scheduling. In *Industrial Scheduling* (J. F. MUTH and G. L. THOMPSON, Eds), pp. 207–219. Prentice-Hall, Englewood Cliffs, NJ.
30. T. J. GIGLIO and H. M. WAGNER (1964) Approximate solutions to the three-machine scheduling problem. *Opns Res.* **12,** 305–319.
31. IBM (1979) *Mathematical Programming System Extended/370 Program Reference Manual.* International Business Machines.
32. L. SCHRAGE (1986) *Linear, Integer, and Quadratic Programming with LINDO*, 3rd edn. The Scientific Press, Palo Alto, Calif.
33. J. L. KUESTER and J. H. MIZE (1973) *Optimization Techniques with FORTRAN.* McGraw–Hill, New York.
34. S. S. PANWALKAR and A. W. KHAN (1976) An ordered flow-shop sequencing problem with mean completion time criterion. *Int. J. Prod. Res.* **14,** 631–635.
35. S. P. BANSAL (1977) Minimizing the sum of completion times of n-jobs over M-machines—a branch and bound approach. *AIIE Trans.* **9,** 306–311.
36. W. SZWARC (1983) The flow-shop problem with mean completion time criterion. *IIE Trans.* **15,** 172–176.
37. D. A. WISMER (1972) Solution of the flowshop-scheduling problem with no intermediate queues. *Opns Res.* **20,** 689–697.
38. S. S. REDDI and C. V. RAMAMOORTHY (1972) On the flow-shop sequencing problem with no wait in process. *Opl Res. Q.* **23,** 323–331.
39. J. N. D. GUPTA (1976) Optimal flowshop schedules with no intermediate storage space. *Nav. Res. Logist. Q.* **23,** 235–243.
40. W. SZWARC (1977) Special cases of the flow-shop problem. *Nav. Res. Logist. Q.* **24,** 483–492.
41. W. SZWARC (1978) Permutation flow-shop theory revisited. *Nav. Res. Logist. Q.* **25,** 557–570.
42. W. SZWARC (1981) A note on the flow-shop problem without interruptions in job processing. *Nav. Res. Logist. Q.* **28,** 665–669.
43. E. F. STAFFORD (1988) An MILP model for the NIQ flowshop. Unpublished working paper, University of Alabama, Huntsville, Ala.
44. E. F. STAFFORD (1983) Evaluating flowshop sequencing heuristics with mixed-integer linear programming. Unpublished working paper, University of South Carolina, Columbia, SC.
45. D. BOOTH and S. TURNER (1987) Comparing selected flow shop scheduling heuristics. Paper presented at the *ORSA/TIMS National Meetings*, St Louis, Mo.
46. S. FRENCH (1987) *Sequencing and Scheduling: An Introduction to the Mathematics of the Job-Shop.* Ellis Horwood, Chichester.
47. W. J. SELEN (1981) A preemptive goal programming for the standard flowshop problem and the flowshop problem without intermediate queues. Graduate course term paper, University of South Carolina, Columbia, SC.
48. J. L. ARTHUR and A. RAVINDRAN (1980) A branch and bound algorithm with constraint partitioning for integer goal programming problems. *Eur. J. Opl Res.* **4,** 421–425.
49. W. J. SELEN and D. D. HOTT (1986) A mixed-integer goal-programming formulation of the standard flow-shop scheduling problem. *J. Opl Res. Soc.* **37,** 1121–1128.
50. W. J. SELEN and D. D. HOTT (1986) New formulation and solution of the flowshop scheduling problem with no in-process waiting. *Appl. Math. Model.* **10,** 246–248.
51. E. F. STAFFORD and R. TYAGI (1987) Computational experience with a MILP model for flowshop scheduling. Paper presented at the *ORSA/TIMS National Meetings*, St Louis, Mo.