IIE Transactions (2000) **32**, 679–686

A simulated annealing approach to mixed-model sequencing with multiple objectives on a just-in-time line

PATRICK R. McMULLEN[1] and GREGORY V. FRAZIER[2]

[1]*College of Business, Department of Management, Auburn University, AL 36849, USA*
E-mail: pmcmullen@business.auburn.edu
[2]*College of Business Administration, Information Systems and Management Sciences Department, The University of Texas at Arlington, Arlington, TX 76019-0437, USA*
E-mail: frazier@uta.edu

Received April 1998 and accepted September 1999

This paper presents a Simulated Annealing based heuristic that simultaneously considers both setups and the stability of parts usage rates when sequencing jobs for production in a just-in-time environment. Varying the emphasis of these two conflicting objectives is explored. Several test problems are solved via the Simulated Annealing heuristic, and their objective function values are compared to solutions obtained via a Tabu Search approach from the literature. Comparison shows that the Simulated Annealing approach provides superior results to the Tabu Search approach. It is also found that the Simulated Annealing approach provides near-optimal solutions for smaller problems.

1. Introduction

Benefits of successful implementations of Just-in-Time (JIT) production systems are well-documented – less inventory, shorter lead times, improved responsiveness, etc. One important part of successful JIT implementation is scheduling the products to be produced. JIT scheduling is the focus of this research. Specifically, this research is concerned with sequencing production of assembly lines where differing products must be made simultaneously, in an intermixed sequence, often referred to as mixed-model scheduling.

In mixed-model production systems, managers would ideally like to sequence the different products as evenly inter mixed as possible, but at the same time not incur an excessive number of setups, or changeovers, due to switching between different products (if setup times are significant). A good sequence of products should have an acceptable level of product intermixing and also an acceptable number of required setups. So in this environment, the sequencing decision becomes a multi-objective problem.

In this research, two specific sequencing objectives are examined: number of required setups and usage rate. A setup is required each time two consecutive items in the production sequence are different products. The usage rate is a measure of the company's ability to keep the schedule level, or evenly intermixed – keeping the raw materials for the different products arriving to the system at as constant a rate as possible. Because JIT systems are concerned with having the right parts at the right place at the right time, sequencing must be done so that the raw materials are introduced into the system at a fairly uniform rate. Miltenburg [1] has presented a metric to measure this usage rate, which was adopted for this research. This metric can be thought of as a surrogate for a firm's ability to produce several items at a uniform rate. For both the number of setups and Miltenburg's usage rate metric, smaller values are desirable.

It should be noted that minimization of setups and usage rate are two of many possible objectives of interest [2]. The majority of research related to this particular effort, however, is most concerned with obtaining sequences providing minimal usage rates: Tamura *et al.* [3], Bard *et al.* [4], Inman *et al.* [5], Xiaobo *et al.* [6], and Miltenburg *et al.* [7] have all made contributions in this area. It is also worth noting that Ghosh and Gagnon [8] as well as Yano and Bolat [9] provide surveys of assembly line sequencing, while Duplaga and Bragg [10] compare heuristics for "smoothing" parts usage. Little research has been done on simultaneous consideration of both usage and number of setups [18].

Production schedules or sequences that provide a strong level of product inter mixing clearly will require more setups. Conversely, fewer setups result in less product inter mixing. In addition to this tradeoff, another issue that complicates the sequencing problem is its combinatorial nature. Typically, an enormous number of possible production sequences exists, even for relatively small problems, so finding the optimal solution is usually impractical. For this type of combinatorial optimization problem, search heuristics have been demonstrated in the literature to be successful. In this research, the heuristic search technique of Simulated Annealing is used to address this multiple-objective sequencing problem. Simulated Annealing differs from most other types of local search heuristic procedures because during the search process, it sometimes replaces a current solution with a relatively inferior solution, to try to avoid getting "trapped" at local optima.

This research presents a Simulated Annealing methodology to solve the mixed-model sequencing problem with the two objectives of minimal setups and minimal usage rates. For a set of sample problems, three different search objective functions are presented to obtain solutions at varying levels of emphasis between the two objectives. The obtained objective function scores are compared to the results obtained from a Tabu Search approach, as well as optimal solution results obtained via complete enumeration. General observations and conclusions are then offered based upon the experimental results.

2. JIT sequencing and combinatorial optimization

2.1. *Problem complexity*

The problem of finding an optimal production sequence using an arbitrary or general performance measure is well known to be of the computational complexity class NP-hard [12]. As such, many heuristic approaches have been developed to solve the sequencing problem. Some approaches have been developed to focus on only one particular performance measure, while others are general enough to be used with any performance measure. The Simulated Annealing heuristic presented in this paper is flexible enough to be used with any performance measure that can be measured directly from the sequence. It also allows multiple performance measures to be used, if the relative priority weightings of the different performance measures are specified.

Before presenting the equations for computing the two performance measures of interest (number of setups and usage rate), the following variables are defined:

U = usage rate of a production sequence;
S = number of setups in a production sequence;
a = number of unique products to be produced;
D_T = total number of units for all products or total demand – also represents number of positions in sequence;
d_i = demand for product i, $i = 1, 2, \ldots a$;
$s_k = \begin{cases} 1 & \text{if setup required,} \\ 0 & \text{otherwise.} \end{cases}$
$x_{i,k}$ = total number of units of product i produced over stages 1 to k, where $k = 1, 2, \ldots, D_T$.

The first performance measure, or objective, is minimization of required setups. The number of setups (S) in a production sequence is:

$$S = 1 + \sum_{k=2}^{D_T} s_k, \tag{1}$$

where k is the index of the position in the sequence. If the product in position k is different from the product in position $k-1$, then a setup is required and $s_k = 1$, 0 otherwise. It is assumed here that an initial setup is required regardless of sequence. It should be noted that the setup times are assumed to be sequence-independent, so the setup time for a product on a machine does not depend on which other product preceded it on that machine. In this study, it is also assumed that the number of setups required is used as a simplifying surrogate for the amount of setup time required. In other words, the total setup time is proportional to the total number of setups.

The second objective is minimization of the usage rate (U), as presented by Miltenburg [1]. The usage rate is:

$$U = \sum_{k=1}^{D_T} \sum_{i=1}^{a} \left(x_{i,k} - k \times \frac{d_i}{D_T} \right)^2. \tag{2}$$

A substantial amount of prior research has shown the general relationship between flexibility and setups requirement to be inversely related (for many different types of applications), as explained in the next section.

2.2. *Efficient frontier*

As previously stated, there is an inherent tradeoff between setups and usage for problems of this type, and there is a desire to find a reasonable level of both. Exploitation of an efficient frontier can provide some additional insight into this problem. In the context here, an efficient frontier is the collection of points determined by the "best" combinations of both setups and usage. Figure 1 below shows the efficient frontier for an example problem where

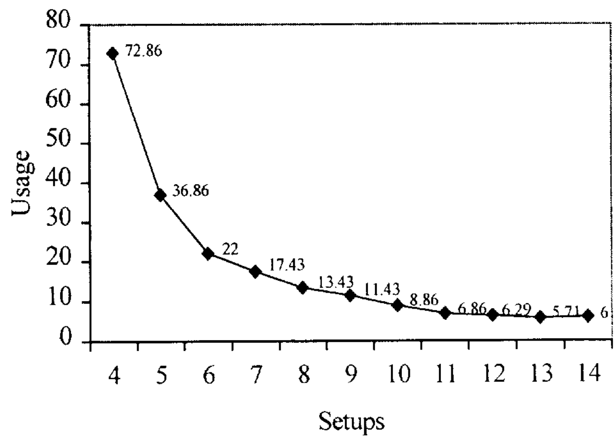

Fig. 1. Efficient frontier of setups and usage.

six units of item A, four units of B, and two units of C and D demanded.

Each point on the frontier represents the minimum usage value for each number of setups – the data comprising this frontier was obtained via complete enumeration. Like efficient frontiers from other applications, this has a convex shape. Of the 1 261 260 possible sequences, only the 11 "efficient" points are shown – the remaining points are located "northeast" of the plotted points, and are considered "inefficient", or dominated. Note from Fig. 1 the inverse relationship between setups and usage, which supports the earlier tradeoff argument. Finding this efficient frontier (or something close to it) without complete enumeration is a challenge, and is left as an opportunity for future research endeavors.

3. Simulated Annealing heuristic

Simulated Annealing is an approach that can provide near-optimal solutions to combinatorial optimization problems. A defining characteristic of Simulated Annealing is that there are occasions when solutions found during the search are replaced with relatively inferior solutions – this in attempt to prevent "local optima trapping." It is assumed here that the reader is familiar with the basics of Simulated Annealing. Kirkpatrick, *et al.* [13], Eglese [14] and Goldberg [15] do provide fundamental descriptions of Simulated Annealing, in addition to informative examples.

In this section, the algorithmic steps of the Simulated Annealing heuristic are explained and insights into the Simulated Annealing search progression are given. It is noted here that for the heuristic, the Simulated Annealing search is guided by the temperature level, T, and the cooling rate, CR – these values essentially dictate the acceptability of solutions found during the search.

3.1. *Heuristic steps*

The steps below describe the Simulated Annealing heuristic for the problem of interest.

Step 1. Initialization.

Initial and final values of the control parameter temperature, referred to as T_I and T_F respectively, are specified. The cooling rate, CR, is specified along with the desired number of iterations (N_{max}) for each level of the current temperature, T. An initial solution is selected from a population of 10 000 randomly generated solutions that has the best composite percentile rank considering both number of setups and usage rate. These composite ranks are determined by finding a percentile rank for both setups and usage for each sequence. These individual percentile ranks are summed, and the sequence with the most desirable "composite" rank is selected as the initial solution. This results in an initial solution providing relatively desirable levels of both setups and usage rate. The objective function value of this initial solution becomes the objective function values for both the current solution (E_C) and best solution (E_B). The iteration counter (N) is set to 1.

Step 2. Objective functions.

The decision-maker chooses a search objective function for Simulated Annealing. Three alternative objective functions were evaluated in this research, to demonstrate how varying emphases can be placed on the two general goals of low number of setups and low usage rate. The general format for the objective functions is:

$$\text{Minimize } E = w_S S + w_U U, \tag{3}$$

where w_S and w_U are the relative weights placed upon the two objectives of number of setups (S) and usage rate (U), respectively. The three objective functions are as follows:

$$\text{Minimize } E_1 = w_S S + w_U U, \tag{4}$$

$$\text{Minimize } E_2 = 3\,w_S S + w_U U, \tag{5}$$

$$\text{Minimize } E_3 = w_S S + 3\,w_U U. \tag{6}$$

The first objective function, E_1, gives equal consideration to both setups and usage. The second, E_2, gives three times the consideration to setups as compared to usage. The third objective function, E_3, gives three times the consideration to usage as compared to setups. Weighting factors are determined as follows:

$$w_S = C/S. \tag{7}$$

$$w_U = C/U. \tag{8}$$

For these weighting factors, C is a constant selected by the user and S and U are the respective setups and usage values of the initial solution obtained via the sampling previously described. The general objective function expressed in Equations (3) and (4) have been constructed such that setups and usage make equal contributions to the objective function.

Step 3. Generate a feasible neighboring solution.

Once the problem has been initialized and an objective function has been chosen, a neighboring solution is generated. This is done through random selection of positions in the sequence and subsequent "swapping". For this research, pairwise swaps are pursued. For pairwise swapping, two unique products are randomly selected and swapped. This new sequence (obtained via the swap) is referred to as the test solution and its objective function value is determined (E_t). It should be noted that other types of solution modifications (such as three-way swaps) were investigated, but experimentation revealed that these other modification approaches did not result in objective function differences significant from the pairwise swaps.

Step 4. Compare test solution with current solution.

If the objective function value of the test solution is greater than the objective function value of the current solution ($E_t > E_C$), proceed to Step 5. Otherwise, if the objective function value of the test solution is less than the objective function value of the current solution ($E_t < E_C$), the test solution will replace the current solution. Then compare this test solution's objective function value with that of the objective function value of the best solution found thus far (E_B). If the objective function value of the test solution is less than that of the best solution found thus far ($E_t < E_B$), then replace the best solution with that of the test solution. Regardless of whether the best solution is replaced with the test solution, proceed to Step 6.

Step 5. Examine Metropolis criterion.

Determine the percent difference between the objective function values of the test solution and the current solution. This difference is denoted by ΔE and is calculated as:

$$\Delta E = 100\left(\frac{E_t - E_c}{E_c}\right). \tag{9}$$

The Metropolis Criterion [16] is then used to determine the probability at which the relatively inferior test solution should be accepted. This probability is as follows:

$$P(A) = \exp(-\Delta E/K_b T). \tag{10}$$

The value K_b is referred to as the Boltzman constant – this value essentially gives the user some control over the probability of inferior solutions being accepted. A random number is then generated in the interval (0,1). If this random number is less than $P(A)$, then the test solution replaces the current solution. Regardless of acceptance status, proceed to Step 6.

Step 6. Increment N.

Increase the increment counter, N, by one. If the value of N is less than or equal to the desired number of iterations for each temperature level ($N_{\max}$), return to Step 3. Otherwise, proceed to Step 7.

Step 7. Adjust temperature.

Adjust temperature by its cooling rate. Mathematically, this is as follows:

$$T = T \times CR. \tag{11}$$

If the new value of T is greater than or equal to the stopping value of T (if $T \geq T_F$) then reset N to one and return to Step 3. Otherwise, stop.

4. Design of experiment

To evaluate the presented methodology, the Simulated Annealing technique was used to solve several test problems contributed by Sumichrast and Russell [17], in addition to some additional larger test problems. These test problems are outlined in the Appendix – problem sets 1, 2, and 3 are from the Sumichrast and Russell publication, and problem set 4 was generated for this research, so that a larger problem could be explored. The objective function values of these solutions were compared to solutions to the same problems obtained via a Tabu Search approach by McMullen [18].

The central research question, then, is as follows:

- Is there any significant difference in objective function performance between the Simulated Annealing and Tabu Search approaches? If so, which approach provides more desirable results?

The secondary research question is as follows:

- Does the Simulated Annealing heuristic provide near-optimal solutions?

To responsibly address these research questions, a fair comparison needs to be made between the two approaches – they need to be compared "on an even playing field". The tables below detail the search parameters for both the Simulated Annealing and Tabu Search approaches. An important issue to note from these two tables is that the total solutions evaluated are equal for the approaches across each problem set, implying that neither technique has an advantage in terms of solutions evaluated. For each problem evaluated, the same initial solution and weighting scheme is used for both the Simulated Annealing and Tabu Search approaches.

The final two columns in Table 1 require some explantion – "Initial $P(A)$" and "Inferiority base". These values essentially determine the probability of inferior solutions being accepted as current. They are therefore used to compute the Boltzman constant. At the initial temperature (T_1) a solution inferior to the current solution by "Inferiority base" percent has an "Initial $P(A)$" percent of being accepted. For example, Problem Set 2 states that a solution 5% inferior to the current solution in terms of the objective function value has a 25% probability of being accepted at T_1.

Table 1. Simulated Annealing search parameters

Problem set	*Cooling rate (%)*	*Iterations for each temperature level*	*Total solutions evaluated*	*Initial P(A) (%)*	*Inferiority base (%)*
1	97	30	3180	50	10
2	97	35	3710	25	5
3	98	40	6400	15	5
4	98.5	45	9685	10	5

Note: For all problem sets, the initial temperature (T_1) is 25 and the final temperature (T_F) is one.

Table 2. Tabu Search parameters

Problem set	*Iterations*	*Sample size for each iteration*	*Total solutions evaluated*	*Tabu list length*
1	1060	3	3180	3
2	1237	3	3711	5
3	1280	5	6400	9
4	1917	5	9685	15

Table 3. Constant values for the Simulated Annealing and Tabu Search Approaches

Problem set	*Constants (C)*	*Initial values for objective function 1*	*Initial values for objective functions 2, 3*
1	1000	2000	4000
2	1000	2000	4000
3	1000	2000	4000
4	10 000	20 000	40 000

Table 2 presents parameter values for the Tabu Search approach. Tabu Search differs from Simulated Annealing in that it is a memory-based approach where a "list" of recent modifications is maintained, and this list determines whether or not a candidate solution becomes current. A detailed description of Tabu Search is beyond the scope of this paper, but Glover offers informative descriptions [11, 19].

A listing of constant values (C) used to determine weighing factors, along with a listing of objective function initial values is provided in Table 3. Also listed are the initial values that the objective functions take on at the start of the Simulated Annealing search procedure.

It is also important to note that for both approaches explored, experimentation was done to determine search parameters that would result in the most desirable levels of objective function values, so that each approach could be shown in its best possible light.

5. Experimental results

Prior to addressing the research questions, the mean number of setups, mean usage rates and mean percentage improvement over the initial solution (obtained via the sampling of 10 000 solutions) are presented in Table 4 for the Simulated Annealing approach, organized by objective function type and problem set.

Recall that objective function 1 gives equal consideration to setups and usage rate. Note from the above table that Objective 2 results in fewer setups at the expense of higher usage rates. Similarly, Objective Function 3 results in lower usage rates at the expense of more setups. In terms of improvement over the initial solutions, it is clear that improvement increases in direct proportion to problem size. The reason for this is because larger problems simply offer much more "room for improvement," due to their combinatorial enormity. Also worth noting with regard to improvement is that objective function 3 generally shows more improvement as compared to objectives 1 and 2. This can be attributed to the fact that reduction in usage rate contributes to sizable improvements in objective function value, due to the weights being "locked in" from the initial solution (which offer higher usage rates).

To address the first research question, ratios of mean Tabu Search objective function value to Simulated Annealing objective function value are presented. Table 5 shows the means and standard deviations of these ratios for each problem set organized by objective function.

This table shows that with the exception of objective function 3 used for problem set 2 (which is in bold), the Simulated Annealing approach generally performs better than the Tabu Search approach. Ratios >1 indicate that objective functions obtained via Tabu Search result in higher objective function values than those obtained via Simulated Annealing, and *vice-versa*. To determine whether these ratio comparisons indicate significant differences in objective function performance, one-tailed t-tests are performed. Table 6 shows t-statistics and associated p-values.

This table shows that in most situations, the superior performance of Simulated Annealing compared to Tabu Search is significant at the $\alpha = 0.02$ level. Exceptions to this are problem set 1, objective function 3, and problem set 2, objective functions 2 and 3 (where, of course,

Table 4. Mean setups and usage rates for the Simulated Annealing approach

Problem set	*Objective function 1*			*Objective function 2*			*Objective function 3*		
	Setups	*Usage*	*Percentage improvement (%)*	*Setups*	*Usage*	*Percentage improvement (%)*	*Setups*	*Usage*	*Percentage improvement (%)*
1	12.44	15.83	8.26	10.44	22.04	6.04	14.78	12.67	21.48
2	15.22	32.13	5.21	13.44	39.81	4.84	17.56	29.57	12.72
3	47.00	609.96	55.83	42.67	746.34	47.61	58.78	431.93	67.20
4	201.67	59481	76.70	199.33	76686	65.72	211.67	39064	87.66

Table 5. Ratio comparison of objective function values for Simulated Annealing and Tabu Search solutions

Problem set	*Objective function 1*	*Objective function 2*	*Objective function 3*
1	1.028 (0.033)	1.029 (0.033)	1.006 (0.024)
2	1.011 (0.012)	1.005 (0.024)	**0.998 (0.014)**
3	1.308 (0.217)	1.262 (0.141)	1.244 (0.285)
4	1.501 (0.042)	1.533 (0.117)	1.424 (0.031)

Table 6. T-statistics (and associated p-values) of ratio comparisons

Problem set	*Objective function 1*	*Objective function 2*	*Objective function 3*
1	2.55 (0.0172)	2.64 (0.0149)	0.75 (0.2374)
2	2.75 (0.0125)	0.62 (0.2747)	−0.43 (0.6602)
3	4.26 (0.0014)	5.57 (0.0003)	2.57 (0.0166)
4	20.66 (0.0012)	7.89 (0.0078)	23.69 (0.0009)

problem set 2, objective function 3 shows Tabu Search slightly outperforming Simulated Annealing – although this difference is not significant).

The second research question addresses the optimality (or near-optimality) of the Simulated Annealing approach. To address this research question, complete enumeration was used to find the optimal solution to several problems from problem set 1, objective function 1 (small problems). These optimal solutions were compared to solutions to the same problems obtained via Simulated Annealing. The percentile performance of the Simulated Annealing solutions was then obtained.

Table 7 shows that the objective function values of the Simulated Annealing search approach the optimal objective function values obtained via complete enumeration – a reasonable argument for near-optimal solutions obtained from the simulated Annealing approach.

Table 7 also shows the number of solutions that were found (during enumeration) to be superior to the Simulated Annealing approach in terms of objective function. This information was then converted into percentiles. For example, the Simulated Annealing approach for problem C, was found to be superior to 99.978 070 18% of all other enumerated solutions in terms of objective function. The mean percentile for these problems above was found to be statistically equivalent to the 100th percentile ($t = -1.01$, $p = 0.18$). These results provide evidence of the near-optimality of the Simulated Annealing approach. It is desirable to extract this type of information for all problems evaluated, but the combinatorial complexity only permits enumerating all solutions for the smaller problems.

6. Concluding comments

The search technique of Simulated Annealing has been presented to find production sequences when the objectives of setup minimization and minimization of usage rate are present. The Simulated Annealing approach generally outperforms the Tabu Search approach in terms of objective function value, and also provides near-optimal solutions to the smaller problems.

The issue of weighting is one that requires consideration. Here, weighting was engineered such that both setups and usage rate made equal contributions to the base objective function (objective function 1). These weights were then adjusted to vary the emphasis on these two objectives. The point that should be realized is that weighting schemes here are very general and in practical applications, these weights would need to be engineered in accordance with the practitioners goals.

Exploitation of the efficient frontier (as presented earlier) is another way to address the problem at hand. "Mapping" the best combination of the setups and usage rate objectives essentially presents the "best" solutions without considering the issue of weighting. Finding the efficient frontier is the challenge here. Beam Search is one approach that can be used to find, or at least approach, the efficient frontier when there are multiple objectives to be considered. De *et al.* [20] and Nair *et al.* [21] have used the Beam Search approach to address multi-objective problems. The Beam Search approach could also be used to address the multi-objective problem addressed here and the authors consider it an opportunity for future research.

Table 7. Comparison of Simulated Annealing solutions to optimal solutions obtained via enumeration

Problem set 1 problem	*Simulated annealing solution*	*Optimal solution*	*Possible solutions (Permutations)*	*Solutions superior to SA solution*	*Simulated annealing percentile (%)*
B	2000.00	2000.00	116 280	0	100.000 000 00
C	2000.00	1961.50	930 240	204	99.978 070 18
D	1916.61	1910.66	16 279 200	36	99.999 778 86
E	1742.54	1710.92	1 396 755 360	112	99.999 991 98
F	1735.80	1731.35	2 993 047 200	8	99.999 999 73

Acknowledgement

The authors are grateful for the constructive feedback they received from the Departmental Editor and the anonymous referees. This feedback was very helpful in the improvement of the paper.

References

[1] Miltenburg, J. (1989) Level schedules for mixed-model assembly lines in just-in-time production systems. *Management Science*, **35**(2), 192–207.

[2] Bolat, A. (1994) Sequencing jobs on an automobile assembly line: objectives and procedures. *International Journal of Production Research*, **32**(5), 1219–1236.

[3] Tamura, T., Long, H. and Ohno, K. (1999) A sequencing problem to level part usage rates and work loads for a mixed-model assembly line with a bypass subline. *International Journal of Production Economics*, **61**, 557–564.

[4] Bard, J., Dar-El, E. and Shtub, A. (1992) An analytic framework for sequencing mixed-model assembly lines. *International Journal of Production Research*, **30**(1), 35–48.

[5] Inman, R. and Bulfin, R. (1991) Sequencing JIT mixed-model assembly lines. *Management Science*, **37**(7), 901–904.

[6] Xiaobo, K. and Ohno, A. (1994) A sequencing problem for a mixed-model assembly line in a JIT production system. *Computer and Industrial Engineering*, **27**, 71–74.

[7] Miltenburg, J., Steiner, G. and Yeomans, S. (1990) A dynamic programming algorithm for scheduling mixed-model just-in-time production systems. *Mathematical and Computer Modeling*, **13**(3), 57–66.

[8] Ghosh, S. and Gagnon, R. (1989) A comprehensive literature review and analysis of the design, balancing and scheduling of assembly systems. *International Journal of Production Research*, **27**, 637–670.

[9] Yano, C. and Bolat, A. (1989) Survey, development and applications of algorithms for sequencing paced assembly lines. *Journal of Manufacturing and Operations Management*, **2**, 172–198.

[10] Duplaga, E. and Bragg, D. (1998) Mixed-model assembly line sequencing heuristics for smoothing component parts usage: a comparative analysis. *International Journal of Production Research*, **36**(8), 2209–2224.

[11] Glover, F. (1990) Tabu search: a tutorial. *Interfaces*, **20**(4), 74–94.

[12] French, S. (1982) *Sequencing and Scheduling: An Introduction to Mathematics of the Job-Shop*, Ellis Harwood Ltd, Chichester, UK.

[13] Kirkpatrick, S., Gelatt, C.D. and Vecchi, M.P. (1983) Optimization by simulated annealing, *Science*, **220**(4598), 671–679.

[14] Eglese, R.W. (1990) Simulated annealing: a tool for operational research. *European Journal of Operational Research*, **46**, 271–281.

[15] Goldberg, D.E. (1989) *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, MA.

[16] Metropolis, N., Rosenbluth, A., Rosenbluth, N., Teller, A. and Teller, E. (1953) Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, **21**, 1087–1092.

[17] Sumichrast, R.T. and Russell, R.S. (1990) Evaluating mixed-model assembly line sequencing heuristics for just-in-time production systems. *Journal of Operations Management*, **9**, 371–390.

[18] McMullen, P.R. (1998) JIT sequencing for mixed-model assembly lines with setups using tabu search. *Production Planning and Control*, **9**(5), 504–510.

[19] Glover, F. (1993) A user's guide to tabu search. *Annals of Operations Research*, **41**(1), 3–28.

[20] De, P., Ghosh, J.B. and Wells, C.E. (1992) Heuristic estimation of the efficient frontier for a bi-criteria scheduling problem. *Decision Sciences*, **23**(3), 596–609.

[21] Nair, S.K., Thakur, L.S. and Wen, K. (1995) Near optimal solutions for product line design and selection: beam search heuristics. *Management Science*, **41**(5), 767–785.

Appendix

Sample problems

Problem set 1. (number of each product type)

Problem	Product 1	Product 2	Product 3	Product 4	Product 5
A	20	0	0	0	0
B	16	1	1	1	1
C	15	2	1	1	1
D	13	4	1	1	1
E	10	5	2	2	1
F	8	7	2	2	1
G	6	6	5	2	1
H	5	5	5	3	2
I	5	4	4	4	3
J	4	4	4	4	4

Problem set 2. (number of each product type)

Problem	Pr 1	Pr 2	Pr 3	Pr 4	Pr 5	Pr 6	Pr 7	Pr 8	Pr 9	Pr 10
A	20	0	0	0	0	0	0	0	0	0
B	11	1	1	1	1	1	1	1	1	1
C	10	2	1	1	1	1	1	1	1	1
D	9	3	1	1	1	1	1	1	1	1
E	8	4	1	1	1	1	1	1	1	1
F	7	5	1	1	1	1	1	1	1	1
G	6	5	2	1	1	1	1	1	1	1
H	5	5	3	1	1	1	1	1	1	1
I	4	4	4	2	1	1	1	1	1	1
J	2	2	2	2	2	2	2	2	2	2

Problem set 3. (number of each product type)

Problem	Pr 1	Pr 2	Pr 3	Pr 4	Pr 5	Pr 6	Pr 7	Pr 8	Pr 9	Pr 10	Pr 11	Pr 12	Pr 13	Pr 14	Pr 15
A	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	40	40	8	1	1	1	1	1	1	1	1	1	1	1	1
C	35	35	10	5	5	1	1	1	1	1	1	1	1	1	1
D	30	30	15	10	5	1	1	1	1	1	1	1	1	1	1
E	25	25	20	15	5	1	1	1	1	1	1	1	1	1	1
F	20	20	20	15	15	1	1	1	1	1	1	1	1	1	1
G	20	20	15	15	10	6	6	1	1	1	1	1	1	1	1
H	15	15	15	10	10	10	10	5	4	1	1	1	1	1	1
I	15	15	10	10	10	10	10	10	4	1	1	1	1	1	1
J	7	7	7	7	7	7	7	7	7	7	6	6	6	6	6

Problem set 4. (number of each product type)

Prob.	Pr 1	Pr 2	Pr 3	Pr 4	Pr 5	Pr 6	Pr 7	Pr 8	Pr 9	Pr 10	Pr 11	Pr 12	Pr 13	Pr 14	Pr 15	Pr 16	Pr 17	Pr 18	Pr 19	Pr 20
B	105	105	105	105	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
F	35	35	35	35	35	35	35	35	35	35	15	15	15	15	15	15	15	15	15	15
J	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25	25

Biographies

Patrick McMullen is an Assistant Professor at the Auburn University College of Business, specializing in Operations Management. McMullen received a Ph.D. in Operations Management from the University of Oregon. Prior to that he was an industrial engineer in the food and automotive industries. McMullen has held teaching positions at the University of Maine and Harvard University. His research interests are in applications of search heuristics and he has published in journals such as: *International Journal of Production Research, International Journal of Production Economics, Journal of Productivity Analysis, Production Planning and Control, Journal of the Operational Research Society and European Journal of Operational Research.*

Gregory Frazier is an Assistant Professor of Information Systems and Management Science at the University of Texas at Arlington's College of Business. Frazier received his Ph.D. from Texas A& M University, and is certified as a CFPIM by the American Production and Inventory Control Society. Frazier has held teaching positions at the University of Oregon and Harvard University. His research interests are in cellular manufacturing and production scheduling. He has published in journals such as *International Journal of Production Research, Journal of Operations Management, Business Horizons, Journal of Productivity Analysis, Production and Inventory Management Journal and International Journal of Production Economics.*

Contributed by the Feature Applications Department