A SIMPLE METHOD FOR THE MULTI-ITEM, SINGLE-LEVEL, CAPACITATED SCHEDULING PROBLEM WITH SETUP TIMES AND COSTS

STUART J. ALLEN
JACK L. MARTIN
School of Business, Penn State-Erie, Erie, PA 16563

EDMUND W. SCHUSTER
Welch's, Concord, MA 01742

In this article, we describe the development of a "home-grown" system for scheduling packaging lines at Welch's, a major food manufacturer. The system is simple in that it requires no specialized math programming knowledge or software but is based on intuitively appealing rules which achieve feasible schedules while accounting for holding and setup costs. Schedules are produced in an Excel 5.0 spreadsheet using Visual Basic macros. We cannot claim that the results are optimal, but we will provide evidence later that they are competitive with the most sophisticated mathematical procedures currently available. At any rate, no known methods can guarantee a priori that optimal or even feasible solutions will be found for scheduling problems of this type.

The software, which we describe in subsequent sections, was developed at the Center for Process Manufacturing, a partnership with APICS and Penn State-Erie, and sponsored by Welch's. The Center is dedicated to helping process industries rationalize decision making by translating academic studies into practical applications.

In keeping with the goals of the Center for Process Manufacturing, our purpose here is not only to tell the story of a successful scheduling implementation but to provide sufficient detail that others can develop a scheduling tool uniquely suited to their own needs.

THE PRODUCTION SCHEDULING ENVIRONMENT

Some of the factors which the scheduling procedures must address are:

- A make-to-stock manufacturing firm with no stockouts or backorders permitted
- Multi-item, single-level, dedicated production lines with finite capacity
- Setup times and/or costs are non-zero and sequence independent
- Sequencing of multiple items to be produced within a specific time period is not considered
- Safety stocks (buffers) are determined "outside" the scheduling system.

For scheduling problems with sequence dependencies, it may be possible to eliminate these dependencies by grouping items into families. This leads to a hierarchical approach to scheduling and we refer the interested reader to [1].

DATA REQUIREMENTS AND PRECONDITIONING

For the scheduling system we have designed, data collection and preliminary calculations are performed outside of the scheduling module.

Data Requirements

We assume the following data have been collected:

1. Demand forecasts, F(i, t), for each item number i (i = 1, 2, . . . , NITEMS) and each period t (t = 1, 2, . . . , HORIZ).
2. Safety stocks, B(i, t), for each item and each period. Welch's safety stocks are not constant but change by period reflecting dynamic demand patterns. They incorporate both forecast error and bias. See Schuster and Finch [6] for a description of the Welch's buffer computations.
3. Capacity absorption coefficients, A(i), by item. This measures the units of capacity absorbed per unit of output produced for each item.
4. Holding costs, H(i), by item, $/unit/period.
5. Setup cost, SC(i), by item, $/setup.
6. Setup time, ST(i), by item, units of capacity/setup.

7. Production capacity, L(t), by time period, in some convenient time measure.
8. Beginning inventories, I(i, 0), by item. Note that this is the total of all inventory for an item irrespective of how it is allocated—as buffer or toward future demands.

Preliminary Computations

Before invoking the scheduling system, two sets of computations must be carried out using the data described above.

1. Incorporate *net changes* in safety stocks over subsequent periods into the demand forecasts for each item. This is accomplished as follows:

$$D(i, t) = F(i, t) + B(i, t) - B(i, t - 1) \text{ with } B(i, 0) = 0.$$

2. Net out beginning inventories from demands for each item, to obtain final demands, D(i, t).

A complete set of preconditioned data is given in Table 1 for an eight-item, 13-period packaging line at Welch's. We will use these data to illustrate the final results of the scheduling process.

The process easily handles holding and setup costs, setup times and capacity absorption coefficients which vary over items and production capacities that vary by period. In our illustration these remain constant, but this is not to be construed as a limitation on the method. In addition, the scheduling system does not require zero demand over all items in early periods as reflected by the Welch's example in Table 1. These demands simply reflect that we are dealing with a consumer good with associated large inventories which must be netted out prior to the scheduling process.

As an example of a more general set of conditions we have applied the system to the Lucas Workcentre example given in Dixon and Silver [2]. This is a 20-item, 13-period scheduling problem with varying holding and setup costs, capacity absorption coefficients and production capacities. This example contains heavy demands in the early periods but zero setup times. We obtained a feasible schedule for the case of zero setup times with a cost of $6,032, while Dixon and Silver [2] report a cost of $5,944. However, their solution and cost are in doubt since the schedule is not consistent with the original demand data. We have also obtained feasible schedules for

TABLE 1: Final Form of Scheduling Data

Product, i	Demand, D(i,t), units Period, t													Hold Cost, $/unit/per. H(i)	Setup Cost, $/setup SC(i)	Setup Time, hrs. ST(i)	Cap. Absorb., hrs./unit A(i)
	1	2	3	4	5	6	7	8	9	10	11	12	13				
1	0	0	0	0	0	0	0	0	.50	.70	.50	.50	.50	166	200	6	2.5
2	0	0	0	0	0	0	0	.30	.65	.85	.80	.80	.80	166	200	6	2.5
3	0	0	0	0	0	2.40	2.40	2.40	2.40	1.90	2.40	2.40	2.40	166	200	6	2.5
4	0	0	.70	1.45	1.45	1.45	.85	.85	.85	.25	.85	.85	.85	166	200	6	2.5
5	0	0	0	0	0	0	0	0	0	0	0	.65	.65	166	200	6	2.5
6	0	0	0	0	1.00	.85	1.10	1.10	1.10	1.20	1.10	1.10	1.10	166	200	6	2.5
7	0	0	0	0	0	0	0	1.70	1.70	1.35	1.70	1.70	1.70	166	200	6	2.5
8	0	0	0	0	0	0	0	.80	.80	.60	.80	.80	.80	166	200	6	2.5
Capacity, L(t), hrs.	32	32	32	32	32	32	32	32	32	32	32	32	32				
Feas Reqmt R(t),hrs.	75.5	107.5	131.8	154.1	168.0	170.3	173.4	151.5	121.5	94.4	64	32	0				

the Lucas Workcentre data for a variety of assumed setup times.

AN OVERVIEW OF THE SCHEDULING METHODOLOGY

The scheduling problem we face with non-zero setup times is very difficult to solve to optimality. The problem is very easy to pose mathematically as a mixed-integer programming (MIP) formulation (see [7]). However, with our current state of knowledge there is no way to know in advance whether or not an optimal solution will be found. As Trigeiro et al. [7] put it, ". . . one cannot expect to find a fast algorithm to tell whether or not a feasible solution exists, let alone find an optimal solution."

For this reason, most efforts at solving this problem have focussed on heuristic methods combined with math programming. We will not pursue that direction since it requires rather specialized knowledge and software. Instead we have examined the large literature on scheduling with setup costs but zero setup time (see Maes and Van Wassenhove [4] for an excellent survey). Dixon and Silver [2] and Maes and Van Wassenhove [3] are representative of pure rule-based approaches to this problem.

The Dixon-Silver heuristic is conceptually appealing, requires only simple calculations and uses the proven Silver-Meal [5] method of determining economical transfers of future demands, backward in time to current production. Our strategy then was to adapt the Dixon-Silver method to the case of non-zero setup times.

A Summary of Important Module Functions

We first give a brief description of the key subroutines of the scheduling system.

1. **Read data.** Data of the type shown in Table 1 reside in an Excel 5.0 spreadsheet in the form of named ranges. These data are read into one- and two-dimensional arrays in a Visual Basic macro, which is attached to the spreadsheet as module 1.
2. **Compute feasibility requirements for each period.** The feasibility requirement for each period, R(t), is that amount of capacity which must be expended in that period, over and above the immediate demand for that period, to overcome infeasibilities in later period(s). A sticky problem here is that we have no way of knowing in advance how many setups will be required in the final solution.
3. **Satisfy immediate demand and feasibility requirements.** For each period, beginning with period one, immediate demand is first satisfied and the most economical transfers of future demands are then carried out backward in time to the current period, using full or partial lots until feasibility requirements are satisfied. Some or all of these demand transfers may result in cost increases since the first priority is to preserve feasibility. If capacity remains after feasibility is satisfied, only cost-saving transfers are permitted thereafter. If capacity is exhausted before feasibility requirements are met, additional capacity is allocated to that period so that the feasibility requirement can be satisfied. Upon completion of this step, we have a production schedule specifying which items are to be produced and in what quantity for each period. This schedule will not necessarily be feasible. However, we have more transfers to carry out, forward in time, in the next module, and these transfers frequently eliminate infeasible periods.
4. **Improve the solution.** Here cost-saving transfers are carried out forward in time until no further improvements can be found. If earlier infeasibilities are not satisfied in this process, it is not necessarily true that no feasible solution exists. However, as we will see later, this method does find feasible solutions where mixed-integer programming methods have failed.
5. **Report solution results.** Production and inventories by item and period along with costs are written from the Visual Basic arrays into Excel 5.0 spreadsheet ranges. In addition, the report exhibits any additional capacity requirements, by period, when no feasible solution is found.

Feasibility Requirements

Maes and Van Wassenhove [3] presented a simple method of computing feasibility requirements, R(t), for the case of zero setup time. This method recursively computes R(t) backward in time beginning at the horizon:

$$R(\text{HORIZ}) = R(13) = 0,$$

$$R(t-1) = R(t) + \text{Period t Production} - L(t).$$

Now production in period t should include both production and setup time

$$\text{Production}(t) = \sum_{i=1}^{\text{NITEM}} [A(I) * D(i, t) + ST(i)].$$

But if we include every setup, we are assuming a lot-for-lot solution which clearly will overstate the feasibility requirements. This typically leads to overpro-

duction in early periods and large capacity add-ons to the capacity limits. The improvement module may not be able to eliminate these early infeasibilities. Even when it does, the heavy distribution of production early in time cannot be entirely overcome by forward transfers, and holding costs tend to be higher than necessary.

On the other hand, we must be careful not to unduly limit the estimated number of setups. If we understate the feasibility requirements, there will be too little production in early periods and resulting add-ons to capacity in later periods. Late period capacity add-ons are seldom remedied by the improvement module.

Our solution to this dilemma is to have the user intervene in setting the maximum number of setups to be included in the feasibility computations in any period. A search process is then carried out beginning at the largest number of setups to be permitted and decreasing until infeasibilities occur in later time periods or solution cost stops improving. For data in Table 1, the best cost, feasible solution occurs when no more than seven setups are allowed in the feasibility requirement computation. Using a maximum of seven setups and R(13) = 0, the reader can now determine that:

$$R(12) = R(13) + \sum_{i=1}^{8} [A(i) * D(i, 13) + ST(i)] - L(13)$$

$$R(12) = 0 + [22 + 42] - 32 = 32,$$

and

$$R(11) = R(12) + \sum_{i=1}^{8} [A(i) * D(i, 12) + ST(i)] - L(12)$$

$$R(11) = 64.$$

The remaining R(t) follow similarly and are listed in Table 1. We emphasize that setting the maximum number of setups to be counted toward feasibility requirements in no way limits the number of setups that can be used in the remainder of the scheduling procedure.

Satisfy Immediate Demand and Feasibility Requirements

We illustrate the structure of this module with the flowchart in Figure 1. The process of finding the closest future period with non-zero demand deserves further consideration. So long as the feasibility requirements are not completely satisfied in the current period, we must insist that any admissible transfer of production

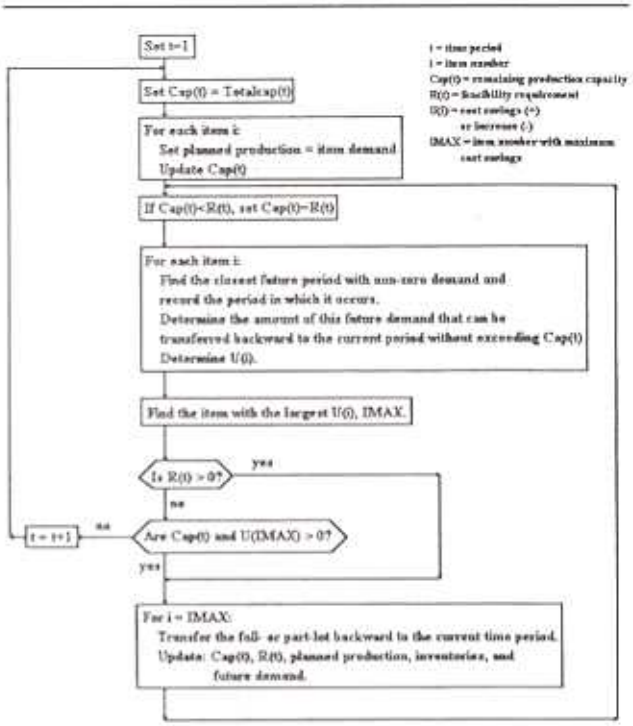

FIGURE 1: Module which satisfies immediate demand and feasibility requirements

from a future period, backward in time to the current period, reduces the feasibility requirement of the current period. Situations can arise in which feasible transfers backward to the current period will not reduce the feasibility requirement but will reduce current production capacity. If these are not prohibited, capacity can be exhausted before the feasibility requirements are satisfied.

To demonstrate an inadmissible transfer, consider the following situation not related to Table 1 data. The current period is t = 1, we have satisfied immediate demand, remaining capacity is ten units and we have a feasibility requirement of five units. Suppose we restrict our example to one specific item, call it item k, and the closest non-zero demand for the item is ten units and occurs in period 3. However, the intervening period 2 has a feasibility requirement of zero units. The situation is summarized in Table 2.

Recall that the feasibility requirement for any period represents production that must occur in that period over and above immediate demand in order to prevent infeasibilities from occurring in future periods. Since period 2 has no feasibility requirement, the set of demands in period 3 are feasible and can be met entirely out of that period's capacity. However, period 1 has a

TABLE 2: Example of an Inadmissible Transfer for Item K from Period 3 into Period 1

	Demand, D(i,t), Units Period, t			
Product, i	*1*	*2*	*3*	...
1	...	...	...	...
⋮	...	...	...	...
k	20	0	10	
⋮	...	...	...	...
R(t), units	5	0	15	...
Cap(t), units	10	30	50	...

non-zero feasibility requirement of five units which means that period 2 cannot meet immediate demand with available capacity. This infeasibility must be absorbed by production in period 1. Now the transfer of ten units of demand from period 3 to period 1 is possible since ten units of capacity are available. However, such a transfer must not be admissible since, if it wins the savings competition among items, such a transfer will eliminate period 1 capacity and not remove the infeasibility for period 2. When this situation occurs, we make this transfer non-competitive by assigning a large negative value to U(i).

In computing savings of permissible transfers, we have adopted the Dixon-Silver method of forming the savings per unit of capacity absorbed. Savings, U(i), can be positive or negative (cost increase) and are computed using Silver-Meal, i.e., savings per span, without regard to capacity. However, because items are in competition for capacity absorption, multi-period transfers are not considered as they would be in the uncapacitated case. Whenever the "winning" item has $U(i) > 0$, i.e., a true cost saving, its average cost and the number of time periods spanned are preserved so that after the transfer they can be used to continue the Silver-Meal average cost per span algorithm.

We will not present here the detailed structure of the equations for the U(i). Four different savings functions are defined: full-lot transfer into an existing setup, full-lot transfer into no setup, and the two corresponding part-lot transfers. Only full-lot transfers into an existing setup present the potential for positive cost savings, $U(i) > 0$. For example, consider time period 5 in Table 1. After producing items 4 and 6 we still have $32 - 2.5\ (2.45) = 25.875$ hours of capacity available. Consider only item 6. The closest future non-zero demand of 0.85 units occurs in period 6. Back shifting the entire lot into period 5 requires only 0.85 $(2.5) = 2.125$ hours of capacity, which is available. The savings would be

$$\text{Savings} = \text{setup cost for period 6} - \text{incr. inv. cost}$$
$$= 200 - .85\ (166) = +58.9.$$

Now Dixon and Silver [2] use savings per unit capacity absorption as the priority index for competing transfers and we follow their lead. Then in period 5,

$$U(6) = +58.9/2.125 = +27.72.$$

This item will win the competition in this period and we must then reduce remaining capacity by 2.125. But the feasibility requirement for period 5 will be reduced by 2.125 + 6.0, the setup hours saved in period 6. This is the means by which infeasibilities can be reduced or eliminated.

Improving the Solution

In the previous module, the primary effort was to reduce feasibility requirements to zero. In order to accomplish this, additional capacity may have been required, typically in the early periods. When additional capacity is needed, in effect, we have an infeasible schedule. Furthermore, feasible or not, the schedule tends to be highly oriented toward early production with relatively high inventory carrying costs. We next search for transfers *forward in time* into periods that have excess capacity. This process will reduce costs and frequently eliminate all additional capacity requirements.

We begin by examining each period, in order, starting with period 2. First we check to see if the period has any remaining capacity. If it does not, we move to the next period. If it does have remaining capacity, then we need to examine each item for potential forward transfers (which cannot exceed remaining capacity). We then choose the item with the largest savings and update costs, production requirements, inventories and capacities.

Suppose we have found a period with remaining capacity. Now suppose that an item has a setup scheduled and there is a positive ending inventory in the *previous* period. We can save carrying costs by producing all possible units of that item in the current period, rather than earlier. If we have sufficient capacity, we may even be able to eliminate an earlier setup cost and time.

For items with no setup scheduled, it may also be possible to realize a savings. If there is an ending inventory for the current period, this must have been produced one or more periods earlier. If there is suf-

ficient capacity to schedule a setup and produce some or all of the inventory in the current period, a potential savings might be realized. We only have to require that the carrying costs saved exceed the additional setup cost. Sometimes it might be possible to completely eliminate the earlier setup, reducing capacity requirements for the earlier period.

After we have checked all periods, we compare the new cost with the original cost. If there has been an improvement, we repeat the entire process and continue to do so until no further cost improvements are found, at which point the procedure ends. The improvement module is summarized in flowchart format in Figure 2.

PACKAGING LINE SCHEDULE AND WHAT-IF ANALYSIS

Final Schedule

The eight-item, 13-period scheduling problem of Table 1 has the feasible schedule given in Table 3. The

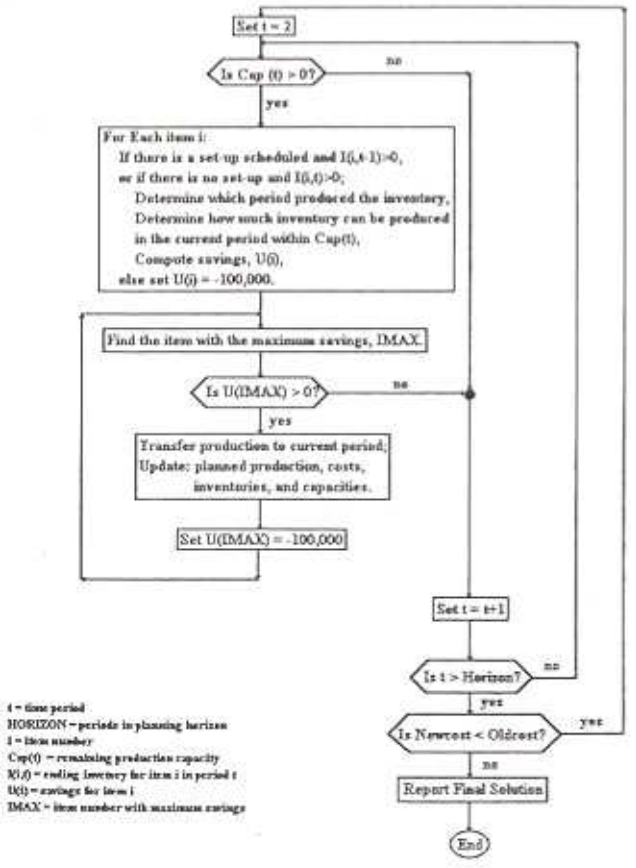

FIGURE 2: Improving the schedule

TABLE 3: Production Schedule for Welch's Packaging Line

COSTS: SET-UP COST = $4400; HOLDING COST = $12998; TOTAL COST = $17398

ITEM #	PERIODS												
	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	0	0	0	0	0	0	0	1.90	0	0.80	0	0
2	0	0	0	0	0	0	0	2.60	0	0	0	1.60	0
3	0	0	0	2.40	0	0	8.40	0	0.70	0	7.20	0	0
4	0	0	2.15	0	4.05	1.40	0	0	0	2.80	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	1.30	0
6	0	0	0	0	2.95	0	0	4.50	0	0	0	2.20	0
7	0	0	0	0	0	3.40	0	0	0	4.75	0	0	1.70
8	0	0	0	0	0	0.80	0	0	3.00	0	0	0	0.80

total cost of the schedule ($17,398) compares very favorably with the best available mixed-integer programming result of $17,724. The MIP computations were carried out on an IBM mainframe using GAMS for problem input and an OSL solver. The solver found an integer valued feasible solution but was not able to improve on our modified Dixon-Silver method even with 100,000 iterations.

Some What-If Investigations

In the cause of succinctness, henceforth, when we refer to the modified Dixon-Silver scheduling method described herein, we will call it simply MODS. The MODS system can also be used to investigate the cost implications of changes in capacity limits, setup times, holding and setup costs, capacity absorption coefficients and demand scenarios. In Table 4 we give two examples of potential cost improvement as setup time is reduced or as capacity is increased. Capacity increases were explored with setup time fixed at 6.0 hours, and for setup time changes, capacity was fixed at 32.0 hours. Each time a parameter is changed it is necessary to carry out a search procedure for the best available feasible solution by varying the maximum number of setups allowed. Normally this involves two or three trials.

TESTS OF CONSISTENCY OF THE SCHEDULING SYSTEM

We now report the results of tests conducted to determine how well the proposed system (MODS) performs. The goal here was twofold: to gain some understanding of how costs compared to the "best available" costs of a mixed-integer programming solution and to uncover any software bugs.

We have been guided by the work done by Trigiero et al. [7] to test their linear programming-based approach to solving capacitated scheduling problems with setup times. Their procedure uses Lagrangian relaxation combined with a "smoothing" heuristic. Tests explored the effects of: level of capacity utilization, setup time level and variability across items, ratio of setup cost/holding cost, problem size (number of items and periods), demand variability, variability of capacity absorption (across items), level and variability of setup and holding costs.

Trigiero et al. [7] measured the gap between final schedule costs and lower bounds on cost obtained in the process of using his algorithm. Average gap over all tests was less than 4%, and this gap is interpreted as a measure of difficulty of obtaining optimal solutions.

Demands were selected from a uniform distribution with the same mean over all items. Varying demand coefficient of variation was found to have no effect on solution gap. Capacity requirements were established by first utilizing EOQ to estimate setup times. Total capacity requirements were then averaged over all periods and divided by the desired capacity utilization factors (75%, 85%, 95%). Using this method, Trigiero et al. [7] found that setup time level and variability across items did not affect the solution gap.

Factors contributing most strongly to an increase in solution gap were: increasing capacity utilization, increasing EOQ time between orders (TBO), and decreasing number of items. Variability of capacity absorption factors across items did not affect solution gap.

TABLE 4: Effect of Reducing Setup Time or Increasing Capacity

Capacity Increase-Hrs	Cost-$	Setup Time-Hrs	Cost-$
0	17,398	6	17,398
2	15,717	5	15,681
4	14,223	4	12,955
6	13,625	3	10,738
8	11,965	2	9,771
		1	9,491

For our testing, we have elected to vary the following factors: capacity utilization (55%, 75%), TBO (1, 3), number of items (8, 20), setup cost level (mean) (450, 4,500) and setup cost variability (range/mean) (0, 1).

Demand

A mean demand of ten units was used for all items with demands drawn from a uniform distribution on the interval (5, 15) with a coefficient of variation of 0.29. This of course results in every item having a non-zero demand in every period, which is not realistic. Trigiero et al. [7] remedied this by randomly selecting some early periods and assigning zero demands. Non-zero demands were then inflated to preserve the desired mean. This was done "in order to simulate . . . [an] increasing trend" over time.

We have taken a somewhat different approach. Data from actual applications at Welch's and reported by Dixon and Silver [2] indicate approximately 45% to 50% of entries in the demand array are zero. We have randomly selected 45% of our demands to be zero and adjusted the non-zero entries upward to preserve the mean of 10.0.

Capacity Absorption and Setup Time

Trigiero et al. [7] found in preliminary tests that varying capacity absorption coefficients did not have an effect on solution gap. They chose to use a coefficient for unity for all items in subsequent tests and we will do likewise. They also found that variability of setup times "had a minor effect on the results." For this reason, we have elected to hold setup time constant at 6.0 units of capacity. While this choice is somewhat arbitrary, it is consistent with the Welch's example and represents a significant fraction (20%) of production capacity for a lot-for-lot schedule at 75% utilization.

Setup and Holding Costs

Choice of an EOQ time between orders implies a (mean) ratio of setup to holding cost for given demand. A TBO of 1 implies a ratio of 5 while TBO = 3 yields a ratio of 45. Holding costs were computed using these ratios and the given mean levels for setup costs (450, 4,500). In the case of variable setup costs, both holding and setup costs were drawn *independently* from uniform distributions with ranges equal to their respective means.

Production Capacity

Once demand and TBO are prescribed, the number of setups can be estimated, and minimum total capacity follows. This is then averaged over all periods and

TABLE 5: Cost Comparison of MODS Method and Best Available MIP Solution

Run Number	TBO	Cap Util Factor%	# of Items	Setup Cost	Setup Cost Var	Hold Cost	Cap	MODS Solution Cost	Best Avail MIP Cost *
1	1	55	8	[450]	1	[90]	194	23,584	23,584
2	1	55	8	4500	0	900	194	256,500	256,500
3	1	55	20	450	0	90	477	64,350	64,350
4	1	55	20	[4500]	1	[900]	477	681,200	681,200
5	1	75	8	450	0	90	142	32,157	28,638
6	1	75	8	[4500]	1	[900]	142	325,012	314,310
7	1	75	20	[450]	1	[90]	350	70,285	68,846
8	1	75	20	4500	0	900	350	680,490	680,130
9	3	55	8	450	0	10	164	19,917	22,013
10	3	55	8	[4500]	1	[100]	164	207,415	226,604
11	3	55	20	[450]	1	[10]	402	48,506	NA
12	3	55	20	4500	0	100	402	495,610	NA
13	3	75	8	[450]	1	[10]	120	19,748	NA
14	3	75	8	4500	0	100	120	215,840	NA
15	3	75	20	450	0	10	295	51,527	NA
16	3	75	20	[4500]	1	[100]	295	522,887	NA

[a] denotes random selection from a uniform distribution with mean a and range / mean = 1.0.
* 50,000 iterations

the result is divided by the capacity utilization factor (55% and 75%). With TBO = 1, setups are lot-for-lot and the capacity attains its largest value. For TBO = 3, capacity is at its smallest value and it was not possible to find feasible solutions using MODS for any capacity utilization factors (CUF) above 75%. This, then, determined the upper value of CUF. As we will see shortly, the MIP solver was unable to find feasible solutions for this combination of TBO and CUF.

Test Design and Cost Results

Our intent was to run a one-half fraction of a 2^5 factorial design to determine how the factors affected the difference between our costs and the best available MIP costs. Based on our experiences with using MIP to solve scheduling problems of this type, best available was taken to be MIP feasible solution cost at 50,000 iterations.

Examination of Table 5 reveals that our original intent did not bear fruit since in six out of the 16 test problems, the MIP solver was unable to find a feasible solution. Run number 15 was taken to 330,000 iterations but no feasible solution was located by the solver. No further improvement was found for the feasible solution in run number 5 at 623,000 iterations. All MODS run times were less than ten seconds using a 33 mhz desktop computer.

Runs numbered 1 to 4 are all lot-for-lot solutions and may well be optimal but the MIP solver was not able to demonstrate this to be the case within the 50,000 iteration limit. None of the other runs yielded lot-for-lot schedules.

The MODS method performs quite well with the worst-case cost penalty of some 12% in run number 5. The other two cost penalties are under 3.5%. There is, of course, no way to assess how well the method performed in the last six runs, but feasible solutions were found, certainly a major advantage over no solution whatever.

REFERENCES

1. Allen, S. J., and E. W. Schuster. "Practical Production Scheduling with Capacity Constraints and Dynamic Demand: Family Planning and Disaggregation." *Production and Inventory Management Journal* 35, no. 4 (1994): 15–21.
2. Dixon, P. S., and E. A. Silver. "A Heuristic Solution Procedure for the Multi-Item, Single-Level, Limited Capacity, Lot-Sizing Problem." *Journal of Operations Management* 2, no. 1 (October 1981): 23–39.
3. Maes, J., and L. N. Van Wassenhove. "A Simple Heuristic for the Multi-Item Single-Level Capacitated Lot-Sizing Problem." *Operational Research Letters* 4, no. 6 (April 1986): 265–273.
4. ———. "Multi-Item Single-Level Capacitated Dynamic Lot-Sizing Heuristics: A General Review." *Journal of Operational Research Society* 39, no. 11 (1988): 991–1004.
5. Silver, E. A., and H. Meal. "A Heuristic for Selecting Lot-Size Quantities for the Case of a Deterministic Time-Varying Demand Rate and Discrete Opportunities for Replenishment." *Production and Inventory Management Journal* 12, no. 2 (1973): 64–74.
6. Schuster, E. W., and B. J. Finch. "A Deterministic Spreadsheet Simulation Model for Production Scheduling in a Lumpy Demand Environment." *Production and Inventory Management Journal* 31, no. 1 (1990): 39–42.
7. Trigiero, W. M., L. J. Thomas, and J. O. McClain. "Capacitated Lot Sizing with Setup Times." *Management Science* 35, no. 3 (March 1989): 353–366.

About the Authors—

STUART J. ALLEN teaches management science and is a member of the Center for Process Manufacturing at Penn State-Erie. He works on design of decision aids for application in manufacturing environments.

JACK L. MARTIN is an independent consultant in the area of production planning and scheduling. He received his PhD in operations management from the Pennsylvania State University. He has published in Operations Research *and* Interfaces. *Jack is a member of APICS, DSI and INFORMS.*

EDMUND W. SCHUSTER is manager of logistics planning at the corporate office of Welch's, Inc., Concord, Massachusetts. An active member of APICS, he currently serves as associate director of the Center for Process Manufacturing at Penn State-Erie. Ed's research interests include building mathematical models for application in industry.