A Case Study in Parallel Unrelated Machine Scheduling: A Heuristic Approach

M.I. Dessouky, University of Illinois at Urbana-Champaign, Urbana, Illinois
Y.M. Dessouky, Arizona State University, Tempe, Arizona
M.M. Dessouky, Pritsker and Associates, West Lafayette, Indiana

Abstract

This article describes a heuristic algorithm that generates weekly production schedules for packaging a number of products on parallel unrelated lines. Various constraints are imposed, including restrictions on the total number of lines that can run simultaneously, product families that can be processed concurrently, total run hours per shift, and available material. Special consideration is given to weekly startup learning coefficients, changeover learning coefficients, changeover times, shutdown hours, and a host of other factors. The objective is to maximize weekly output or to minimize the time to produce weekly demand, whichever is applicable.

Keywords: Machine Scheduling, Heuristic Algorithms, Decision Support Systems, Production Schedules.

Introduction

Objectives. The purpose of this paper is to describe a computerized decision support system developed for scheduling the packaging operation in a food processing plant. The goals of the system are:

1. To expedite weekly scheduling, presently performed manually.
2. To facilitate sensitivity analysis and schedule revision.
3. To test the impact of expanding the system by adding more products, product families, and lines.
4. To permit synchronizing the scheduling of the packaging operation with that of preceding operations.

The Problem. Although the scheduling system described here was developed in a specific context, the approach followed is applicable to problems where a set of jobs is to be scheduled on a number of parallel resources with different productivities, under various constraints. The problem and the algorithm are first presented in a general framework, and then illustrated through the case study.

The production system features product items grouped into families. The operation to be scheduled (packaging) follows a sequence of operations which provide its material inputs. Product quantities are expressed in terms of transportable units of volume, which are converted into metric tonnes for material planning. Metric tonnes of a family are further converted into equivalent tonnes, which are units of material that require the same production capacity of preceding operations regardless of family.

The operation in question (packaging) is performed on a set of parallel machines (lines). A product may be processed (packaged) on more than one line, but a line may handle only one product at a time. The production rate depends on both the product and the line. After a shutdown or changeover, a line starts at a speed lower than normal and gradually accelerates until it reaches its normal production rate after an estimated number of hours.

The time available for processing is limited by the working hours in the week, shutdown, and operator availability. The quantity to be produced is constrained by the demand on the product items in units, and the material supplied by preceding operations in equivalent tonnes. The upper bound on material availability drops after the week's startup, and gradually increases until it reaches its normal level after an estimated number of hours. Some families cannot be simultaneously packaged because they cannot be processed concurrently on preceding operations. Products are assigned priorities that reflect the market need. Furthermore, the number of lines running simultaneously and the number of running hours per shift are restricted by operator availability. Accordingly, the processing time of a product cannot be estimated in advance, since it depends on several factors, including its place in the sequence and the tightness of constraints.

Given the weekly production requirements of all products, the objective is to determine the schedule of all products on all lines in the week, under the prevailing constraints. The criteria for choosing a schedule are:

1. To minimize the number of hours to produce all requirements in the week; or if all products cannot be produced within the week, then
2. To produce the maximum equivalent tonnage in the week over all products.

Literature Review. The problem addressed in this paper is that of scheduling n jobs on m unrelated (nonidentical) machines (processors) in such a manner as to minimize the makespan or to maximize production within a limited time span. Several approaches have been taken to develop exact (optimal) and approximation algorithms for solving this problem under different assumptions. The following algorithms are representative of the literature.

Rothkopf[1] considered the problem with the objective of minimizing the sum of task waiting and processing costs. The waiting costs are linear and are treated with and without discounting. The author developed a dynamic programming algorithm to solve the problem when costs and processing times fall under any of three special cases. These are (1) the waiting cost function is zero; (2) the processing times of a job are inversely proportional to the machine efficiency (the uniform machine case) and the waiting cost function is constant; and (3) the processing times of a job are constant for all machines (the identical machine case) and the waiting cost function is exponential.

Ibarra and Kim[2] studied the computational performance of several heuristic algorithms with the objective of minimizing the makespan. In all these heuristics, a task is scheduled on the processor which minimizes its finishing time. The characteristics of the task to schedule next differ among the algorithms. They include (a) the task index number; (b) the maximum processing time; (c) the minimum processing time; and (d) the minimum finishing time on any machine. All algorithms have a worst case bound of m times the optimal solution during a run time of $O(n)$ for algorithm (a), $O(n \log n)$ for (b) and (c), and $O(n^2)$ for (d).

Lawler and Labetoulle[3] developed a linear programming formulation of the problem when the objective is to minimize the makespan, with preemptions allowed. The algorithm provides optimal schedules with an upper bound of $O(m^2)$ on the number of preemptions.

Davis and Jaffe[4] introduced a heuristic algorithm, with three variations, for scheduling n jobs on m parallel unrelated machines to minimize the makespan. In contrast to the algorithms by Ibarra and Kim, the authors develop a list of jobs for each machine in an order of priority which depends on the efficiency of the machine in processing the job. They show that the makespan obtained by their algorithms ranges between $1.5\sqrt{m}$ to $2.5\sqrt{m}$ times the optimal makespan.

In all the algorithms available in the literature, the following assumptions are made:

1. Jobs (tasks) have known processing times on all machines.
2. A job is not simultaneously run on more than one line.
3. No limit exists on the number of lines running simultaneously.
4. Job setup times, if considered, do not depend on the sequence.
5. No constraints are imposed on shift length, the number of hours of running time in a shift, or the quantity produced in a shift.
6. No machine downtime takes place during the span of time considered.

From the problem statement in the previous section, it can be seen that all these assumptions are violated in the problem addressed. Accordingly, the existing procedures are inadequate for solving this problem and new algorithms have to be devised.

Scheduling System Description

The system may be described through a statement of the factors which influence the selection of an appropriate schedule. These include product and process parameters, constraints on process output, and the objectives to be attained.

Product-related System Parameters. Product parameters pertain either to a family I or to a product item J. These parameters will be specified as the need for them arises.

Process-related System Parameters. The actual process output is determined by the normal capacities or production rates of the machines, restricted by a number of factors. These factors are the time for changeover from one product to another, shutdowns, the drop in production rate after a changeover or a shutdown, and constraints on material availability and total operating hours.

Production Rates

$RP(J,L)$: Normal production rate of product J on line L in units/hr. Equals zero if L does not package J.

Changeover Times and Coefficients

Resetting the packaging line after a changeover from one product to another has two effects on the production rate. The first is a downtime on the machine, called changeover time, required for machine setup. The second is a drop in machine productivity followed by a gradual rise to normal production rates, represented by changeover learning coefficients which are multiplied by production rates. No drop in productivity takes place if the successive products have the same package weight. Shutdowns have a similar effect on productivity, depending on their duration.

$CO(J_1,J_2)$: Changeover time from product J_1 to product J_2.

$CH(J,P)$: Changeover learning coefficient of product J in period P following changeover or shutdown, always ≤ 1. A period P does not necessarily coincide with a shift S.

The effect of shutdowns of packaging lines on the values of changeover coefficients depends upon the duration of the shutdown. If the duration is below a certain threshold value TC hours, the changeover coefficient continues after startup as if no shutdown has occurred. If the shutdown exceeds TC hours, the changeover coefficient returns to its lowest value $CH(J,1)$. The cut-off value TC is estimated from past experience and provided by the scheduler. Idle time on any machine is considered as a shutdown for the purpose of computing changeover coefficients.

Material Startup Learning Coefficients

$MS(D)$: Learning coefficient for material availability in period D after a shutdown of material processing exceeding a given threshold of TM hours, where TM is provided by the scheduler. A period D does not necessarily coincide with a shift S.

These coefficients represent the drop and subsequent rise of the upper limit on material provided by the extrusion and coating operations to the packaging lines. They are multiplied by the normal limits on production in a period in total equivalent tonnes. If the shutdown is less than TM hours, no drop takes place in material availabilty.

Objective Function. First, it is required to complete packaging all product requirements at the earliest time during the week (objective A). Failing to accomplish that, it is required to package the maximum quantity in equivalent tonnes during the week (objective B). Expressed symbolically, it is required to find the schedule that will:

A. Minimize time to produce all products (makespan). That is, $\min \max_{L} CL(L)$

where:

$CL(L)$: Completion time of last job on line L,

$CL(L) = \max_{J} CT(J,L)$,

$CT(J,L)$: Completion time of product J on line L.

B. Maximize total production in equivalent tonnes. That is, $\max \sum_{S} \sum_{L} \sum_{J} QE(J,L,S)$, where:

$QE(J,L,S)$: Scheduled quantity of product J on line L in shift S in equivalent tonnes,

$QE(J,L,S) = QU(J,L,S) * UT(J) * TE(I)$, where I is J's family,

$UT(J)$: Conversion factor from units of J to metric tonnes,

$TE(I)$: Conversion factor from tonnes to equivalent tonnes from family I. The equivalent tonnes of family I are the number of tonnes of a reference family which consume as much of the capacity of the critical preceding operations as one tonne of I,

$QU(J,L,S)$: Scheduled quantity of J on L in S in units,

$QU(J,L,S) = HR(J,L,S) * PR(J,L) * CC(J,S)$ in units, (1)

$HR(J,L,S)$: Scheduled running hours of J on L in S,

$CC(J,S)$ = Changeover coefficient of product J in shift S. It is the average of $CH(J,P)$ of periods P falling in S weighted by the hours of P in S.

$$CC(J,S) = \sum_{P\in S} (CH(J,P) * HP(P,S) / \sum_{P\in S} HP(P,S)).$$

where:

$HP(P,S)$: the number of hours of period P after packaging startup falling in shift S.

Constraints. The constraints are the following:

Time constraints

The number of available hours on a line L in a shift S,

$$\sum_{J} HR(J,L,S) + \sum_{J_1,J_2} CO(J_1 J_2) \leq MH(S) - SH(L,S), \quad (2)$$

for J_1 followed by J_2 on L in S, and where,

$MH(S)$: the maximum number of hours in shift S,

$SH(L,S)$: shutdown hours on line L in shift S.

The completion time of the last product on any line should not exceed the end of the work week,

$$CL(L) \leq WE \qquad \text{all } L, \quad (3)$$

where:

WE : The end of the work week.

Quantity Constraints

$$\sum_{S} \sum_{L} QU(J,L,S) \leq REQ(J), \quad (4)$$

where:

$REQ(J)$: the requirements of product J during the week in units of transportable bulks as given by the planning department.

Product Priority Constraints

Greater emphasis is to be placed on fulfilling the requirements of products with higher priority, $PP(J)$, than products with lower priority. Priority is specified by market requirements.

Packaging Line Constraints

The number of lines running simultaneously (not including setup); dictated by number of line operators available,

$$\sum_{L} LO(L) \leq ML, \quad (5)$$

where:

$LO(L)$: a flag indicating that line L is currently operating (running); 1 = running; 0 = not running,

ML : maximum number of lines that can run simultaneously.

The number of running hours on all lines in a shift, not including changeover time; restricted by the number of line operators available,

$$\sum_{L} \sum_{J} HR(J,L,S) \leq RH(S) \quad (6)$$

where:

$RH(S)$: the maximum number of running hours in shift S.

Material Family Constraints

Families that can run simultaneously; imposed by the demand of certain families for the same machine in a preceding operation, thus preventing their concurrent processing.

$$\sum_{L} \sum_{J \in I_1} AS(J,L) * \sum_{L} \sum_{J \in I_2} AS(K,L) = 0, \qquad (7)$$

where:

$AS(J,L,)$: A flag indicating that product J is currently assigned to L; 1 = assigned, 0 = not assigned,
$I_1 I_2$: Two incompatible families (two that cannot be run simultaneously).

Why is '0' which means 'zero', is italicized, and the preceding '1', which means 'one', is not?

Total equivalent tonnes/shift; dictated by the output of preceding operations,

$$\sum_{J} \sum_{L} QE(J,L,S) \leq ME(S) * SC(S), \qquad (8)$$

where:

$ME(S)$: Maximum equivalent tonnes in shift S,
$SC(S)$: Material start-up learning coefficient for shift S, computed as the average of $MS(D)$ weighted by the hours of period D in shift S,

$$SC(S) = \sum_{D \in S} (MS(D)*HD(D,S)/ \sum_{D \in S} HD(D,S)), \qquad (9)$$

where:

$HD(D,S)$: the number of hours of period D after material startup falling in shift S.

Scheduling Algorithm

Overview. The problem just described is an n-job, m-parallel machine problem with lot sizing, shift divisions, shutdowns, setup times, learning curves and several constraints. The constraints include restrictions on production time, quantity produced, resource (line) availability, and material availability. The objective is to minimize the makespan or to maximize production. As mentioned earlier, there is no existing optimal procedure to solve this problem, and the heuristics reported in the literature have been tested under considerably less restrictive conditions.

The algorithm described here is embedded in a decision support system (DSS) which prepares inputs in the form of a "working file" drawing upon information from a database and inputs from the scheduler. An example of this file is shown in *Table 1*. The algorithm then carries out the procedure described below to produce a schedule which specifies for each shift, the running hours of each line, the quantity produced of each product on each line, the total equivalent tonnes, and the maximum equivalent tonnes (*Tables 2* and *3*). It also summarizes the total production during the week of each product on each line.

Because of constraints on computational capabilities, the algorithm has to be limited to single-pass heuristic procedures. Accordingly, several scheduling rules were explored which aim at either attaining system objectives, or meeting its constraints. In spite of the existence of two objectives, it may be seen that maximizing production within a limited period of time tends to minimize the time required to produce a fixed quantity. Therefore, while the two objectives are not identical, they are largely compatible. We chose rules that maximize production per unit time as an attempt to reach both objectives. The scheduling rules, given in the next section, were applied in the selection of a product-line combination.

Scheduling Rules. The rules are classified according to whether they aim at maximizing equivalent production or meeting constraints.

To maximize equivalent production:

Line Selection (L):

1. Highest loading rank, $LR(L)$. In this case, ~~of~~ a higher-rank, line is one which has a higher demand (in hours) on its capacity by the products required during the week. The loading rank is estimated by the scheduler. The rationale is that since the total number of lines simultaneously operating is restricted, lines with higher loading rank should not be kept idle. It is interesting to note that the commonly applied opposite heuristic, assigning products to the line with the lowest demand, is justified only when there is no such restriction.
2. Highest production rate $PR(J,L)$ for the given J.
3. Idleness before starting. This choice will allow at least part of changeover to take place during idle time.

Product Selection (J):

1. Highest equivalent output rate, $PR(J,L)*UT(J)*TE(I)$.

After
Fig 1
Pg
Example

Table 1
Input Data
(Working File)

Time parameters:

Number of Shifts = 15   Length of Shift $LS(S)$ = 8.0 hours, all S
Maximum Total Available hours in shift $MH(S)$ = 23 hours, all S

Starting and Changeover Information:

Threshold material shutdown duration TM = 8 hours
Startup Time segment D = 8 hours
Startup coefficients $MS(D)$ = 0.85, 0.95, 1, 1, ...;
for segments 1, 2, 3, 4,...
Threshold shutdown duration for changeover coefficients TC = 8 hours

$$\text{Changeover time} = \begin{cases} \text{0.5 hours for additional or stopping of coupon} \\ \text{2.0 hours for other changes} \end{cases}$$

Changeover time segment P = 8 hours
Changeover coefficients $CH(J,P)$ = 0.71, 0.86, 0.93, 1, 1,...;
for segments 1, 2, 3, 4, 5,...and all products J.

Production Line Informations:

Number of lines available for use = 4
Maximum number of lines running simultaneously ML = 3
Shutdown hours $SH(L,S)$, = 0 hours all L and S
Loading rank: 3, 1, 2, 4 for lines 1, 2, 3, 4, respectively

Product Information:

Number of Products to be scheduled = 8   Number of families = 3

Family	Product	Priority	Pack.Wt (lbs)	Units to Equiv. Tonnes	Required (Units)
(I)	(J)	$PP(J)$	$PW(J)$	$UT(J)*TE(I)$	$REQ(J)$
A	$A1$	2	50	0.04032	2,016
A	$A2$	1	30	0.02520	8,000
A	$A3$	3	10	0.04536	46,800
B	$B1$	6	50	0.03228	2,016
B	$B2$	7	30	0.02268	9,000
B	$B3$	8	10	0.04536	10,800
C	$C1$	4	5	0.04536	21,800
C	$C2$	5	2.5	0.01750	75,648

Families A and B are incompatible

Production Rates PR (J, L):

	Product J							
Line L	A_1	A_2	A_3	B_1	B_2	B_3	C_1	C_2
1	0	0	0	0	0	0	0	960
2	0	0	510	0	0	510	510	0
3	0	0	280	0	0	280	280	0
4	550	1100	0	550	1100	0	0	0

Maximum equivalent tonnes per shift $ME(S)$ = 464; all S
All tolerances = 0.

Table 2
Results of Inputs in Table 1 Using Line Route Algorithm

			Monday			Tuesday			Wednesday			Thursday			Friday			Total
Line	Prod		1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	
01		Hrs	7	7	7	7	7	7	7	7	7	7	8	4.8**				
	C2	Qty	4,771*	5,635	6,115	6,518	6,720	6,720	6,720	6,720	6,720	6,720	7,680	4,608				75,648
02		Hrs	8	8	8	8	8	8	8	~~8~~ 8	~~8~~ 8	8	8	7.8	4.1	8	8	
	A3	Qty	2,897*	3,509	3,794	4,080	4,080	4,080	4,080	4,080	4,080	4,080	4,080	3,960				46,800
	B3	Qty													1,495	3,213	3,656	8,364
03		Hrs	8	8	8	8	8	8	8	~~8~~ 8	~~8~~ 8	8	1.9**		4.1	7.4		
	C1	Qty	1,590*	1,926	2,083	2,240	2,240	2,240	2,240	2,240	2,240	2,240	521					21,800
	B3	Qty													821	1,615		2,426
04		Hrs											5.1**	6.0	6.0	6.0	7.9	
	A2	Qty											4,014	3,986				8,000
	A1	Qty												504	1,512			2,016
	B1	Qty													831	1,185		2,016
	B2	Qty														2,320	6,680	9,000
Total Operating Hours			23	23	23	23	23	23	23	23	23	23	23	18.6	14.2	21.4	15.9	
Total Equivalent Tonnes			287.0	352.2	373.6	400.7	404.3	404.3	404.3	404.3	404.3	404.3	444.3	381.1	192.8	309.8	317.3	
Maximum Equivalent Tonnes			394.4	440.8	464	464	464	464	464	464	464	464	464	464	464	464	464	

*Quantities are rounded to the nearest digit **Hours are rounded to the nearest tenth of an hour

Table 3
Results of Inputs in Table 1 Using Product Route Algorithm

			Monday			Tuesday			Wednesday			Thursday			Friday			Total
Line	Prod		1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	
01		Hrs		5.1**	7	7	7	7	7	7	7	7	8	8	5.7			
	C2	Qty		3,508*	5,306	5,990	6,440	6,720	6,720	6,720	6,720	6,720	7,680	7,680	5,444			75,648
02		Hrs	8	8	8	8	8	8	8	~~8~~ 8	~~8~~ 8	8	8	7.8**	6.2	8	8	
	A		2,897*	3,509	3,794	4,080	4,080	4,080	4,080	4,080	4,080	4,080	4,080	3,960				46,800
	B3	Qty													2,260	3,374	3,732	9,366
03		Hrs	7	8	8	8	8	8	8	~~8~~ 8	~~8~~ 8	8	2.9**		2.3	4.9		
	C1	Qty	1,392*	1,884	2,046	2,220	2,240	2,240	2,240	2,240	2,240	2,240	800					21,800
	B3	Qty													463	971		1,434
04		Hrs	8	1.9**									4.1	1.2	6.0	8	1.8	
	A2	Qty	6,248*	1,752														8,000
	A1	Qty											1,618	398				2,016
	B1	Qty												92	1,924			2,016
	B2	Qty													838	6,425	1.737	9,000
Total Operating Hours			23	23	23	23	23	23	23	23	23	23	23	17	20.2	20.9	9.8	
Total Equivalent Tonnes			352.0	350.1	358.6	390.5	399.4	404.3	404.3	404.3	404.3	404.3	396.5	322.8	323.4	342.8	208.7	
Maximum Equivalent Tonnes			394.5	440.8	464	464	464	464	464	464	464	464	464	464	464	464	464	

*Quantities are rounded to the nearest digit **Hours are rounded to the nearest tenth of an hour

2. Same package size as last product on line.

Lot Sizing:

1. Run product on line until all requirements are produced or week has ended.

To meet constraints:

Product Selection (J):

1. Highest Priority, $PP(J)$
2. Same family as product just finished or currently being processed on another line.

Other constraints mentioned in the "Constraints" section are observed in the determination of the controllable variables: $HR(J,L,S)$.

Because of the implicit conflict among the rules, we provided two mechanisms to allow flexibility in their application:

1. A product-first option and a line-first option in the selection of the product-line combination.
2. Tolerances that permit overriding particular rules if such overriding promises a better schedule.

Overall Procedure. An overview of the procedure is given in *Figure 1.* In step 1, variables are initialized, including the current time, which is set at zero. If there are no more products to schedule, then the procedure produces the results and stops (steps 2 and 12). Steps 3, 4, and 5 select a product-line combination. Following the product route, the highest priority product is identified and tentatively assigned to the most efficient line for it. For the line route, the line with the greatest load is selected and tentatively allocated to the highest priority product which it can process.

The product-line combination selected is then checked to determine whether it violates any production constraints, which include family compatibility, number of available lines, and number of simultaneously running lines. If any of these constraints is violated, the current product-line combination is discarded, and a new one is selected (steps 6, 7, and 8). Otherwise, the product is assigned to the line and, after accounting for changeover, is scheduled for processing on the line, and the production quantity and time are determined (step 9). Tests are included in this step to ensure the satisfaction of constraints on the number of processing hours and the equivalent tonnes produced in each shift. If the end of the week is reached, the procedure produces the results and stops (steps 10 and 12), otherwise, it updates the current time to the end of processing (step 11), and cycles back to step 2.

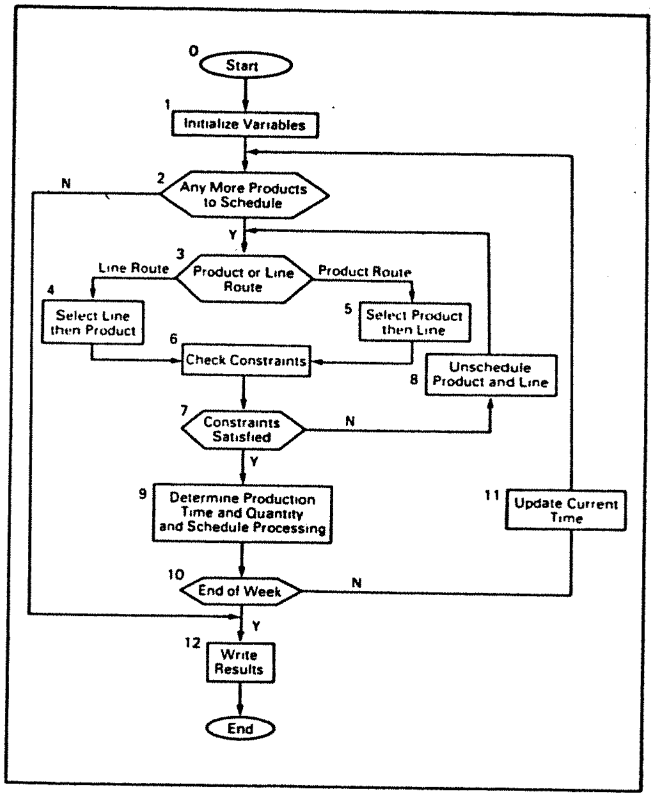


Figure 1
Overview of Procedure

An event-oriented algorithm was designed to apply this procedure, updating the current time ($TCUR$) to the next event upon the completion of an activity. Events are of three types: end-of-processing, end-of-changeover, and end-of-shutdown.

A flow chart of the overall scheduling algorithm is shown in *Figure 2.* As the diagram shows, step 1 reads and initializes the variables. Step 2 determines if there are any remaining products to be scheduled. If all products have been scheduled, the algorithm summarizes and produces the results and stops. Otherwise, it proceeds by selecting a product and an available line (step 4), which is explained in further detail in *Figure 3* and will be discussed later.

If no product and line combination is found, the procedure advances current time $TCUR$ to the next earliest event in step 8. If a combination is found, the completion of changeover is scheduled in step 6. Then, a search for an idle line is conducted in step 7. If a line is found idle, the procedure returns to step 2.

Otherwise $TCUR$ is advanced to the earliest event, step 8. If $TCUR$ is currently at the end of the week, the algorithm writes the results and stops. The

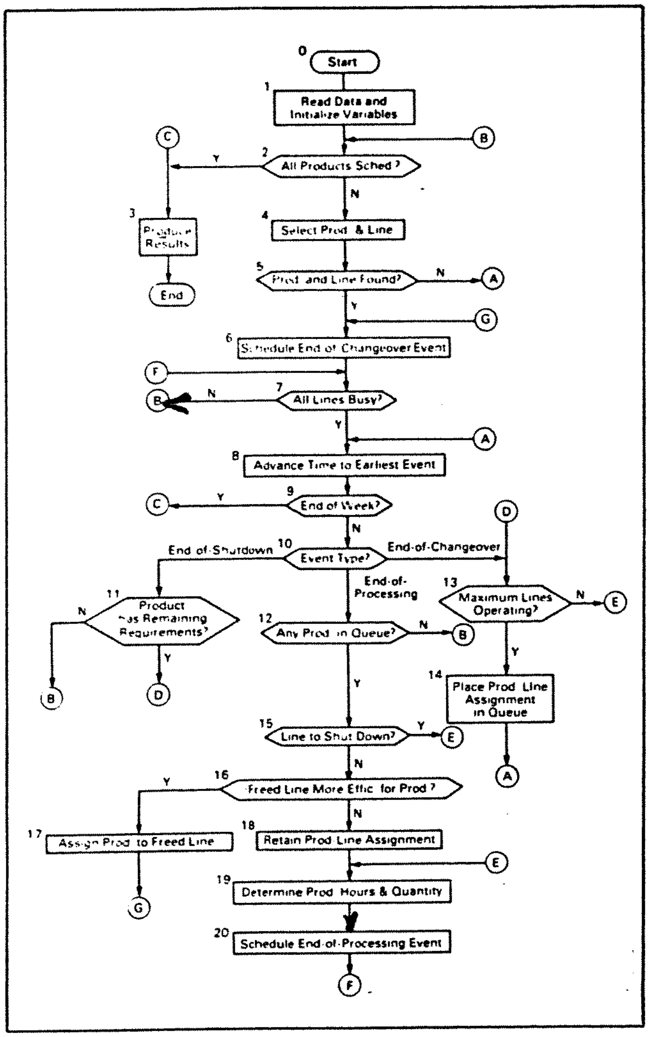


Figure 2
Overall Scheduling Algorithm

type of event which *TCUR* was advanced to is then determined, step 10.

If an end-of-changeover event has been realized, a check is made to find if the maximum allowable number of concurrent processing lines has already been reached, Eq. (5), in step 13. If it has, the product-line combination is placed in a queue, step 14, and the algorithm returns to step 8; otherwise, it proceeds to step 19.

If an end-of-shutdown event has been realized, a check is made for that product which was on the same line before shutdown occurred. The procedure determines whether the week's demand for that product was fulfilled, i.e., Eq. (4) became an equality. If the requirements were fulfilled, the algorithm returns to step 2; otherwise, it proceeds to step 13.

If an end-of-processing event has been realized, a check is made to determine whether a product-line combination is waiting in the queue. If none is waiting, the procedure returns to step 2. Otherwise, step 15 determines if that line will be shut down during the current shift. If it will, the algorithm jumps to step 19.

Steps 16, 17, and 18 determine if the freed line is more efficient for the product than the line that was in the queue. If the freed line is more efficient, the product is scheduled on that line, and the algorithm returns to step 6. However, if the line which was placed in the queue is more efficient, the algorithm proceeds to step 19.

At step 19, the actual production schedule is determined. This step is flowcharted in further

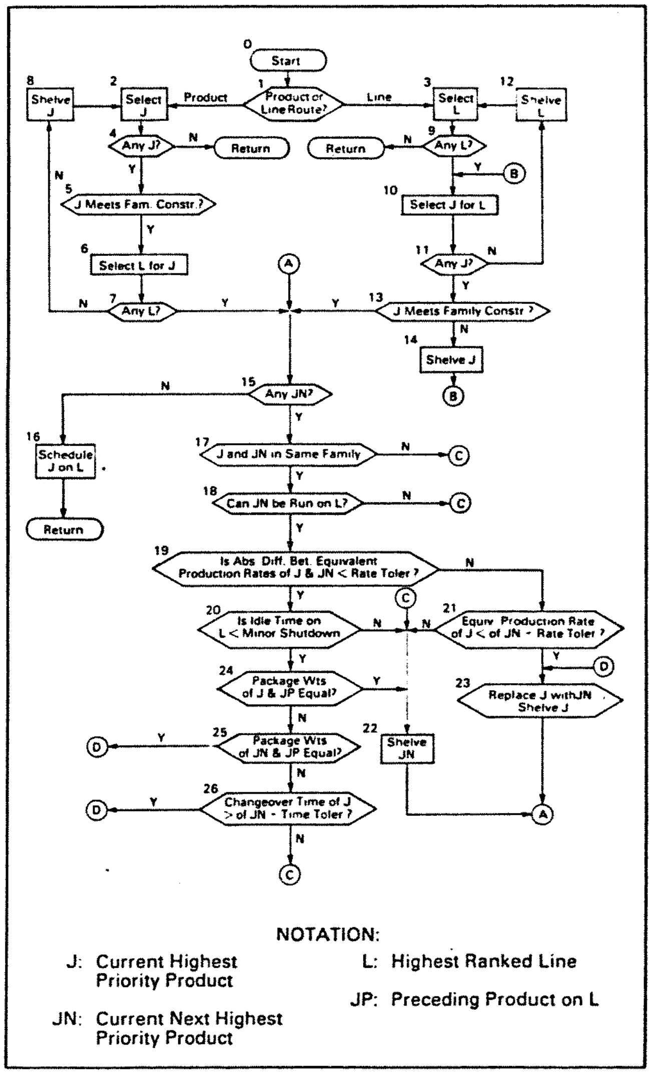


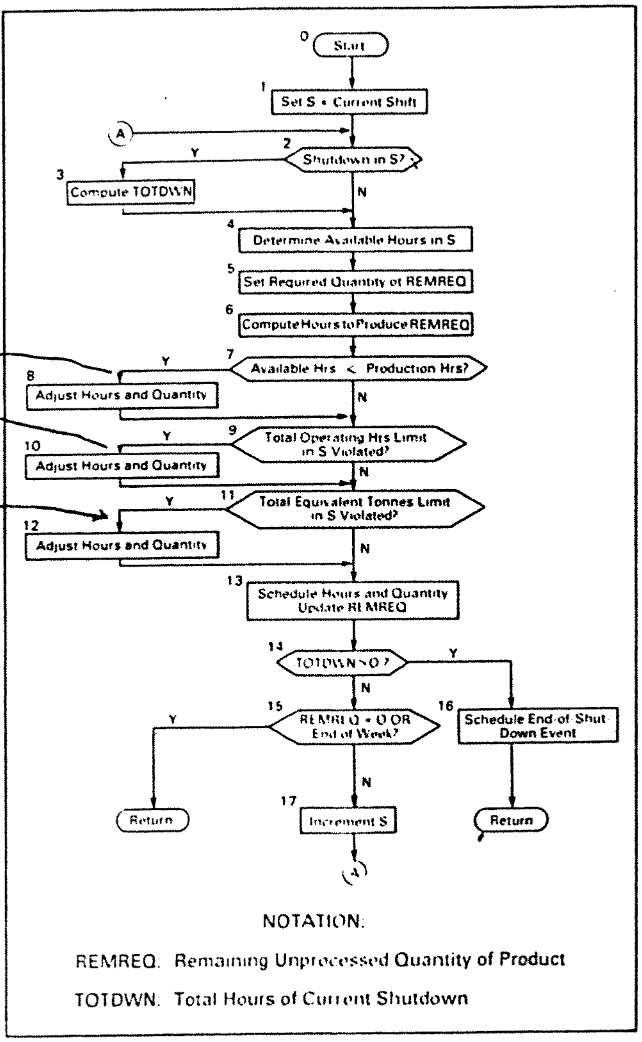
Figure 3
Procedure to "Select Product and Line"

detail in *Figure 4*. The algorithm then schedules a completion of processing event for the current product and returns to step 7.

In conclusion, the algorithm will terminate when all demand has been fulfilled for the current week or *TCUR* reaches the end of the week, Eq. (3).

Product Line Selection Procedure. A discussion of the procedure to select a product line combination is presented below. As represented by *Figure 3*, step 1 determines whether a highest priority product option or highest ranked line option is to be pursued, as designated by the user.

The product option (route) selects the product J with the highest priority $PP(J)$, step 2. If the selected product does not satisfy the family constraint, Eq. (7), or if no line is found for it, steps 4-7, it is shelved and the algorithm returns to step 2 to seek the next highest priority product.



NOTATION:

REMREQ: Remaining Unprocessed Quantity of Product

TOTDWN: Total Hours of Current Shutdown

Figure 4
Procedure to "Determine Production Hours and Quantity"

>> ength of these arrows should be the same as

The line route selects the line L with the highest loading rank $LR(L)$, step 3. An attempt is then made to identify the highest priority available product, step 10. If no product with unscheduled quantities can be packaged on L, the line is shelved and the algorithm returns to step 3 to seek the next highest ranked line. Otherwise, it determines whether the selected product P satisfies the family constraint. If P does not, the algorithm returns to step 10 to find the next highest priority product.

If no product line combination was found in the above steps, the algorithm returns to step 8 in the overall algorithm, *Figure 2*. Otherwise, the selected product, J, becomes a candidate for scheduling on the selected line, L. A check is then made to verify that alternative J is the most desirable available product for the selected line.

Step 15 searches for an alternative product, JN. If no alternative product exists, schedule J and L and return. If some JN is found, it checks if JN belongs to the same family as J and if JN can also be packaged on the selected line. JN is shelved when the above two conditions are not satisfied, and the algorithm then returns to step 15.

If an alternative challenging product JN exists, it is tested against J to determine if it has a higher net equivalent production rate. In this particular system, the net rate is determined by three factors in the given order: (a) the equivalent production rates $ER(J,L) = PR(J,L) + UT(J) * TE(I)$, for J and JN; (b) production loss due to changeover learning coefficients which depend on shutdown times and the package weights of the products; and (c) production loss due to changeover time. User specified tolerances on equivalent production rates and changeover times are employed in this comparison in order to provide the user with the flexibility to trade off product priority against production rate. The test starts with step 19 as follows.

The algorithm takes the absolute difference of the equivalent production rates $ER(J,L)$ and $ER(JN,L)$. If this difference is less than the user specified tolerance, it proceeds to step 20; otherwise, it checks whether $ER(J,L) < ER(JN,L)$ + rate tolerance (step 21). J is replaced with JN only if the above is true. The algorithm then returns to step 15.

At step 20, a check is made on the idle time for the selected line L. If the idle time is greater than the shutdown limit, then no improvement can be made on the changeover coefficient by replacing *J* with *JN*. Therefore, *JN* is shelved, and the algorithm returns to step 15. Otherwise, a check is made to determine whether *JN* has a larger changeover coefficient (through testing package weights) or a smaller changeover time, steps 24, 25, and 26. *J* is replaced with *JN* when either of the above conditions is satisfied and the algorithm then returns to step 15.

Determination of Production Hours and Quantity. This procedure is flowcharted in *Figure 4*. Step 1 identifies the current shift, *S*. Step 2 then determines if any shutdown will occur in *S*. If one will, it calculates the total downtime for the selected line during that shutdown, *TOTDWN*, in step 3 and proceeds to step 4; otherwise, step 3 is skipped.

At step 4, the number of available hours to be scheduled in shift *S* is calculated. The algorithm then sets the production quantity equal to the remaining unprocessed quantity, *REMREQ*. Step 6 computes the number of production hours to process *REMREQ* using Eq. (1). If the available hours are less than the production hours as expressed in Eq. (2), then the production hours for shift *S* are set equal to the available hours, and the quantity produced is adjusted accordingly.

In step 9 the algorithm checks if the total number of operating (running) hours for all lines in *S* exceeds the user-specified limit *RH*(*S*) of Eq. (6). If it does, the production hours for shift *S* is adjusted along with the production quantity, step 10.

The next constraint to be checked is the total equivalent tonnes for all families, Eq. (8), in step 11. If this exceeds the user-specified limit for the shift *S*, *ME*(*S*), an appropriate adjustment is made to the production quantity, and hours, step 12.

Step 13 then schedules the production hours and quantity, and *REMREQ* is reduced by the production quantity. The algorithm then checks if *TOTDWN* is greater than zero, step 14. If it is, the alorithm schedules the end-of-shutdown event (step 16) and returns to the main procedure.

Otherwise, the algorithm determines whether *REMREQ* became zero or production hours for the selected line, *L*, have reached the end of the week, step 15. If any of the above conditions are true, the algorithm returns; otherwise, it increments the shift *S*, step 17, and returns to step 2.

Example

Production System Overview. The products are dry food items divided into families, with each family representing a unique blend, or recipe, of raw materials. Three families are currently being produced, with plans calling for diversification into new families. Each product item represents a particular package size within a family. There are presently 11 items, to be increased up to 30 items.

Upon arrival, the new materials are stored in bunkers or tanks. They are then weighed, mixed, processed, extruded, dried, coated and packaged. The last operation is the focus on this study. Products are packaged into cartons or bags, depending on content weight. Cartons and bags are assembled into larger units for transportation. Demand and production are expressed in terms of these units.

Four packaging lines are available, two additional lines are being installed, and more are contemplated. A product may be packaged on more than one line concurrently, but a line may handle one product at a time. Furthermore, a line is designed to accommodate a limited range of sizes. Because of the learning effect, a line reaches its normal output rate 24 hours after the end of a shutdown or a changeover.

Each shift is staffed by a crew of operators. A crew is trained to run all lines. Staffing requirements, however, limit to a maximum of 23 the total number of run hours on all lines in a shift. The plant currently runs 8-hour shifts, three shifts a day, five days a week, to be increased to seven days a week if necessary.

The material available for packaging is limited by the output of two preceding operations: extrusion and coating. Because of the learning effect on these operation, the upper bound on total equivalent tonnes of material for all families reaches its normal level 16 hours after the week's startup. Since only two coating machines are available, and one of them processes two families, only one of these two families can be processed, and hence packaged, at the same time.

The input data are given in the working file of *Table 1*. A detailed explanation of the line route algorithm is followed by a brief description of the product route procedure.

The Line Route. The following example illustrates the scheduling algorithm presented earlier.

The input data are given in *Table 1*. A detailed explanation of the line route is followed by a brief explanation of the product route. As shown in *Table 1*, the order of the lines by descending loading rank is 2, 3, 1, 4. The results of the line route algorithm are presented in *Table 2*. Note that the maximum equivalent tonnes in shifts 1 and 2 are 394.4 and 440.8, respectively, and that this maximum returns to its normal level 464 in shift 3. This reflects the effect of the startup learning coefficients.

After initializing and checking for products to schedule (steps 1 and 2 in *Figure 2*), the algorithm selects line 2 with the highest loading rank *LR* for scheduling, and searches the product file for the most desirable available product. Product *A*3 is selected because it has the highest product priority, 3, of all products which can be packaged on line 2. These selections correspond to step 4, *Figure 2*, and are detailed in steps 3, 9-11, 13, 15 and 16 in *Figure 3*. Further tests indicate that there are no challenging products since no other product in family *A* can be packaged on line 2. These tests are represented by steps 15, 17, 18, and 22 in *Figure 3*.

Since the current time is at the beginning of the week (*TCUR* = *O*), changeover is assumed completed over the shutdown period preceding the shift and an end-of-changeover event is scheduled at time zero, according to step 6, *Figure 2*. Since all lines are not busy (step 7), the algorithm proceeds to select another line. Line 3 is selected because it has the second highest *LR*. The available products that can operate on this line are *B*3 and *C*1. *C*1 is selected because of its higher product priority, and the cycle is repeated with the selection of line 1 with the third highest *LR*. Product *C*2 is scheduled on line 1 because it is the only product that can operate on this line.

The only available line for scheduling at this point is line 4. The available products for this line are *A*1, *A*2, *B*1, and *B*2. Products from family *B* are eliminated from consideration because they will violate the material family constraint, Eq. (7), (*A*)3 has been chosen for line 2). Also, *A*2 is selected over *A*1 because of its higher product priority.

As all lines are now busy, time is advanced to the earliest event (step 8, *Figure 2*), which is the end of changeover for *A*3 on line 2 at time zero. The constraint on the maximum number of operating lines (3) expressed in Eq. (5) is tested in step 13 and found not exceeded. Accordingly, the weekly requirements of 46,800 are scheduled (step 19). The details of this step are shown in *Figure 4* and proceed as follows. Steps 1 through 6 determine the number of available hours in shift 1 by Eq. (2), and the number of hours needed to produce the required quantity (46,800). Steps 7 through 12 choose the minimum of the two times and check the relevant constraints on time and quantity. Step 13 then computes the scheduled hours of *A*3 in shift 1 as 8, determines the scheduled quantity from Eq. (1) as 2,897, and updates the remaining requirements. Steps 2 through 13 are repeated for each shift until all the week's requirements are satisfied. After passing the tests of steps 14 and 15, the algorithm returns to the overall procedure, *Figure 2*, at step 20, where the end-of-processing event is scheduled to occur at time 95.8 (hour 7.8, shift 12).

The next earliest event is the end of changeover of *C*1 on line 3 at time zero. The cycle is repeated for this combination and an end-of-processing event is scheduled at time 81.9 (hour 1.9, shift 11). Similarly, an end-of-processing event is scheduled for *C*2 on line 1 at time 92.8 (hour 4.8, shift 12). However, an end-of-processing event is not scheduled for *A*2 on line 3 since the number of lines operating is at its maximum (3), as given by Eq. (5). The *A*2-line 4 combination is placed in a queue pending the release of a line, as shown by steps 13 and 14, *Figure 2*. The algorithm then advances *TCUR* to the earliest event, which is the completion time of *C*1 on line 3 at 81.9, as depicted in steps 8 through 10. *A*2 is now removed from the queue, assigned to line 4, and an end-of-processing event is scheduled at time 94 (hour 6, shift 12). This is shown by steps 12, 15, 16, and 18 through 20.

Now, the only available line for scheduling is line 3. Product *B*3 is selected because it is the only product that can operate on that line. The product is placed in a queue for the same reason mentioned above. *TCUR* is then advanced to 92.8, the completion time of *A*2 on line 4. At this time, *B*3 cannot be started because that would violate the material family constraint. Therefore, *B*3 remains in the queue.

The algorithm then attempts to match a product for the available line 4. Product *A*1 is selected because it has the highest priority and does not violate the family constraint. A changeover event is then scheduled at *TCUR* equal to 92.8, when line 4 is released by the completion of *A*2. When changeover is completed, production of *A*1 begins. *TCUR* is then advanced to time 95.8 when production of *A*3 on line 2 has ended.

Since *B*3 can be produced on line 2 as well as line 3, it is placed in a queue for packaging on line 2. Production of *B*3 begins on line 2 when *A*1 is completed on line 4 at time 99.9. Line 2 is selected over line 3 because it has a higher production rate for *B*3 as represented by steps 16 and 17. Since the weekly requirements cannot be met on line 2 by the end of the week, the algorithm schedules the unfulfilled requirements of *B*3 on line 3 to begin at time 99.9, (hour 3.9, shift 13) and end at time 111.4 (hour 7.4, shift 14).

At *TCUR* equal to 99.9, the only available products are *B*1 and *B*2. Both can only be packaged on line 4. *B*1 is completed at time 107, changeover to *B*2 begins and processing continues until all *B*2's requirements have been met at time 119.9. The algorithm terminates when all requirements are produced as shown by steps 2 and 3. Note that the upper bound on the number of hours per shift (23), was reached before the upper bound on the equivalent tonnes per shift. This occurred because lines 1 and 3 were operating for most of the shifts producing products *C*2 and *C*1, respectively, at a low output rate in equivalent tonnes. This was precipitated by the high demand on *C*2, *C*1.

The Product Route. The final schedule produced by the product route of the algorithm for the inputs given in *Table 1* is shown in *Table 3*. As the diagram shows, the product-first selection rule gives *A*2 the top priority requested by the user. This priority was ignored for over 10 shifts by the line route because of the low ranking of line 4, where it can be produced. However, the sequence of all other products remains the same on all lines, and the total requirements are also produced without shortages during the same makespan.

Implementation

The Implementation Environment. The algorithm was designed as part of a larger information system that included the master production schedule, material storage requirements, labor availability and productivity, and a host of other information. The larger information system is accessed through a DEC PDP-11 minicomputer. The programming language chosen for the algorithm was FORTRAN because of its availability and portability. Due to limitations on the addressable memory of the PDP-11, a single-pass heuristic algorithm was designed. The algorithm was first programmed for the CDC Cyber 170-175 and then adapted for the PDP-11.

In order to apply the algorithm in a user-friendly environment, the analysts designed a decision support system (DSS) to prepare the input (working) and output files, and interact with the larger information system. The DDS, described in Reference 5, consists of a scheduling database (a component of the larger system's database) and a data management procedure. It was designed as an interactive system that enables the user to create the working file by extracting information from the database and making the additions and modifications required for a week's production schedule.

The Performance of the Algorithm. During the period the algorithm was tested, 8 to 11 products were scheduled on four lines. The results discussed below are for 11 products. It took an experienced scheduler about two working days to manually develop a schedule, and several hours to revise the schedule as changes in product requirements or system availability and capability took place. These manual schedules were efficient with no apparent room for improvement. However, increases in the number of products and lines were expected to increase scheduling time while reducing schedule quality.

To use the scheduling algorithm, the user first creates the working file with the interactive program, taking three to five minutes on either the Cyber or the PDP-11. The execution time, which included loading, was less than 0.25 seconds on the Cyber and about 15 seconds on the PDP-11. The solutions on the two machines were expectedly similar and their quality was generally good, producing on the order of 10-20% more equivalent tonnes per week than schedules produced randomly. Because the algorithm is a single-pass heuristic, it is not surprising that some of the schedules produced were slightly worse than the manual schedules. Computer schedules were always within 5% of manual schedules, and could be easily refined manually. Revising the schedule requires editing the working file, and running a revised file is not different from running an original file.

Concluding Remarks

The algorithm represents an attempt at developing a schedule for a multiple parallel unrelated machine problem, complicated by a number of constraints and factors. The algorithm relies on heuristics which aim to achieve the stated objectives and meet the imposed constraints. A product route (select product then line) and a line route (select line then product), were provided. Since the schedule produced by each route is data-dependent, the scheduler will always have the option of attempting both routes and choosing the better schedule.

It is interesting to note that problem definition and algorithm development evolved as a result of an interactive effort between analyst and scheduler. Several of the scheduling rules attempted to emulate the scheduler's thought processes. As the study progressed, initial loosely-stated formulations gradually gave way to a more precise problem definition, which was general enough to accommodate the anticipated changes in the system. This formulation helped to generate heuristics which produced more satisfactory schedules.

Based on experience with implementing the algorithm, the analysts may be able to enhance the algorithm through any of the following embellishments:

1. Testing additional heuristic rules for possible adoption.
2. Mixing the line and product routes, choosing whichever route is most suitable when a product-line combination is selected.
3. Developing a computed line loading rank which may change as products are scheduled.
4. Developing a generic version of the algorithm that may be applicable to a broad range of scheduling situations.

Acknowledgement

The authors acknowledge the invaluable suggestions of John W. Nowak and Rick Olson, University of Illinois at Urbana-Champaign, both toward the development of the system and the writing of this manuscript, and to the anonymous referees for their helpful comments.

References

1. M.H. Rothkopf. "Scheduling Independent Tasks on Parallel Processors", *Management Science*, Volume 12, No. 5, 1966, pp. 437-447.
2. O.H. Ibarra, C.E. Kim. "Heuristic Algorithms for Scheduling Independent Tasks on Nonidentical Processors", *J.ACM*, Volume 24, No. 2, 1977, pp. 280-289.
3. E.L. Lawler, J. Labetoulle. "On Preemptive Scheduling of Unrelated Parallel Processors by Linear Programming", *J.ACM*, Volume 25, No. 4, 1978, pp. 612-619.
4. E. Davis, J.M. Jaffe. "Algorithms for Scheduling Tasks on Unrelated Processors", *J.ACM*, Volume 28, No. 4, 1981, pp. 721-703.
5. J.W. Nowak, M.I. Dessouky. "A Decision Support System for Parallel Unrelated Machine Scheduling under Various Constraints," Technical Report, University of Illinois, Urbana, Illinois, 1986.

Author(s) Biography

M.I. Dessouky is an Associate Professor of Industrial Engineering in the Department of Mechanical and Industrial Engineering at the University of Illinois at Urbana-Champaign, Urbana, Illinois. He received a B.Sc. in Mechanical Engineering at Cairo University, Egypt, a M.S. in Industrial Engineering at Purdue University, and a Ph.D. in Industrial Engineering at Ohio State University. Dr. Dessouky's current research activities cover multiple criteria decision making, scheduling, and integrated quality systems. He has published articles in the *IIE Transactions, Journal of Operations Research, Management Science, SIAM Journal of Computing*, the *Computer Journal*, and the *Journal of Engineering for Industry*. He is a senior member of IIE and a member of ORSA, TIMS, and Alpha Pi Mu.

Yasser M. Desssouky is currently a graduate student in Industrial Engineering at Arizona State University, Tempe, Arizona. He received a B.S. in Industrial Engineering at the University of Wisconsin-Madison in 1984. Mr. Dessouky has worked as a Systems Analyst at Pritsker and Associates, W. Lafayette, Indiana, and is a member of IIE and Alpha Pi Mu.

Maged M. Dessouky is a Senior Systems Analyst at Pritsker and Associates, W. Lafayette, Indiana. He received his B.S. in Industrial Engineering at Purdue University in 1984, and is currently pursuing a M.S.I.E. degree at the same university. Mr. Dessouky is a member of IIE, Alpha Pi Mu, Omega Rho and Phi Kappa Phi.