MATHEMATICAL FORMULATION: A PROBLEM IN DESIGN

A. Alan B. Pritsker, Arizona State University
Lawrence J. Watters, The RAND Corporation
Philip M. Wolfe, Arizona State University

February 1968

P-3790

22

MATHEMATICAL FORMULATION: A PROBLEM IN DESIGN

A. Alan B. Pritsker, Arizona State University

Lawrence J. Watters, The RAND Corporation*

Philip M. Wolfe, Arizona State University

ABSTRACT

In this paper, the relationship between formulation and design is examined. The main aspects of mathematical formulation are the development of definitions of variables and the functional relationships of the variables required to model the situation under study. There are many ways to attack these two aspects since there can be many mathematical formulations for the same problem description. The design of an efficient mathematical formulation is an activity analogous to other engineering design activities. In this paper the design aspects of mathematical formulation are discussed and an example illustrating the design considerations is presented.

Two major aspects in mathematical formulation involve the defining of the variables to be included in the formulation, and the developing of the relationships of the variables to form objective functions and constraints. Thus, the formulation of a problem in mathematical terms involves the creation and evaluation of alternatives for definitions and relationships.

In this paper, it will be assumed that the problem to be solved has been specified. It is only necessary to translate the problem description into an efficient mathematical form which is suitable for

---

*Any views expressed in this paper are those of the authors. They should not be interpreted as reflecting the views of The RAND Corporation or the official opinion or policy of any of its governmental or private research sponsors. Papers are reproduced by The RAND Corporation as a courtesy to members of its staff.

solution. The word efficient implies the existence of alternative mathematical forms and a value system for selecting among the alternative forms. For this paper, the measure of efficiency will be the size of the problem in terms of the number of unknowns and constraints. An alternative measure of efficiency would be the time to obtain a solution. A demonstration of the design aspects involved in mathematical formulation will be given through the vehicle of an example. Alternative formulations for the specified problem will be given. The design of the alternative mathematical formulations represents a real engineering challenge.

The remainder of this paper is organized as follows: (1) statement of the problem to be formulated; (2) definitions and relationships for a first design; (3) a redesign in order to reduce the number of unknowns and the number of constraints; (4) a refinement to the redesign; (5) generalizations to a multiproject scheduling problem; and (6) conclusions.

Before proceeding with the problem description and alternative formulations, it is pertinent to point out that the content of this paper is such that advanced mathematical concepts are not required for its reading. The symbolism included in the paper is complex, however all manipulations are strictly algebraic. This further emphasizes the fact that mathematical formulation is concerned with design.

Statement of the Problem

As a vehicle for exploring the design aspects of mathematical formulation, a simple, one-machine sequencing* problem with delay costs

*"Scheduling" is often used as the problem descriptor. Since sequencing implies ordering from which start times can be obtained, i.e. a schedule, the terms are closely related.

[3, 14] will be used. This problem involves the sequencing of a given set of jobs on one machine in such a manner as to minimize the total cost associated with jobs exceeding given due dates. It is assumed that the processing time for each job is known with certainty. If a job is completed before its due date, then no penalty cost is assessed. This problem is a special case of the m-machine n-job scheduling problem, which in turn is a subproblem of the general network scheduling problem when limited resources are available.

A considerable amount of research has been expended on job sequencing and scheduling problems. Several excellent survey papers have been published in this area [15, 19]. A review of the literature dealing with job sequencing is not relevant (the bibliography does include the references), since the main concern of this paper is the treatment of the method used in formulating such problems, not in the solution of such problems. The formulations presented in this paper, however, are different from those currently given in the literature.

In order to obtain a feel for the sequencing problem considered, a small example will first be presented. The example involves six jobs. The input information concerning the six jobs is presented in Table I. The formal definitions of the input information are given below:

g_i = due date of job i (a period number after which penalties are assessed);

d_i = processing time to perform job i (in periods); and

p_i = penalty cost per period that job i is late.

These definitions will be used in all the formulations to be presented in this paper.

One method for sequencing the jobs is by their due dates. That is, the job with the earliest due date is scheduled first, and so on.

Table I

DUE DATES, PROCESSING TIMES AND PENALTIES FOR SIX-JOB, ONE-MACHINE EXAMPLE

Job Number	Due Date	Penalty/ Period Late	Job Duration
1	2	$5.00	5
2	4	4.00	4
3	8	2.00	3
4	12	1.00	5
5	13	7.00	2
6	17	1.00	7

Sequencing jobs by smallest due date yields the sequence and delay costs as shown in Table II. For comparison purposes, the optimal sequence [14] and associated delay costs are shown in Table III. Even with this small example, it is seen that the minimal cost solution is increased by more than 50 per cent when sequencing is done by due date as opposed to an optimal procedure. (This difference in cost has nothing to do with the mathematical formulation but does provide the motivation for obtaining a formulation from which the optimal sequence can be obtained).

First Design

In examining the one-machine sequencing problem, it is seen that for each job there is a sequence number denoting the order in which the job could be performed in relation to the other jobs. Thus, if a position within the sequence is defined, the one-machine sequencing problem appears similar to the assignment problem, and it is only necessary then to assign the jobs to positions in such a manner as to minimize the

Table II

SEQUENCE BASED ON DUE DATE FOR
SIX-JOB, ONE-MACHINE EXAMPLE

Sequence Number	Job Number	Due Date	Penalty/ Period Late	Job Duration	Completion Time	Delay Cost
k	i	g_i	p_i	d_i	T_i	C_i
1	1	2	$5.00	5	5	$ 15.00
2	2	4	4.00	4	9	20.00
3	3	8	2.00	3	12	8.00
4	4	12	1.00	5	17	5.00
5	5	13	7.00	2	19	42.00
6	6	17	2.00	7	26	18.00
					Total	$108.00

Table III

OPTIMAL SEQUENCE FOR SIX-JOB, ONE-MACHINE EXAMPLE

Sequence Number	Job Number	Due Date	Penalty/ Period Late	Job Duration	Completion Time	Delay Cost
k	i	g_i	p_i	d_i	T_i	C_i
1	2	4	$4.00	4	4	$ 0.00
2	1	2	5.00	5	9	35.00
3	5	13	7.00	2	11	0.00
4	3	8	2.00	3	14	12.00
5	6	17	2.00	7	21	8.00
6	4	12	1.00	5	26	14.00
					Total	$69.00

total delay cost. In order to approach this problem mathematically, the first step is to define the unknown of the problem. Based on the observation that the problem can be considered as an assignment problem, the definition that comes to mind for the unknown is

$$x_{ik} = \begin{cases} 1 & \text{if job } i \text{ is performed in the kth position} \\ 0 & \text{otherwise.} \end{cases}$$

At this point in the design process, we have arbitrarily viewed the problem as an assignment problem and selected a definition or, if you like, designed a variable which will enable us to portray mathematically the one-machine sequencing problem as an assignment problem. The examination of the sequencing problem in these terms is based on the scheduling work that has been performed by Wagner [21] in which he defined the unknown in a similar fashion. This building on another researcher's formulation or definition is directly analogous to the procedure used by a design engineer. With this definition of the unknown, x_{ik}, the constraints of the one-machine sequencing problem can be formulated, and again they are analogous to the constraints in the assignment problem. Since each job can only be assigned to one position, we have

$$\sum_{k=1}^{n} x_{ik} = 1 \qquad i = 1,2,\ldots.n. \qquad \text{Eq. 1.}$$

Since every position can only have one job assigned to it, we also require

$$\sum_{i=1}^{n} x_{ik} = 1 \qquad k = 1,2,\ldots.n. \qquad \text{Eq. 2.}$$

Equations 1 and 2 and the 0-1 conditions imposed on the variables represent the constraints necessary in this formation for the one-machine sequencing problem. The ability to rapidly design constraining equations was due to our understanding of the basic assignment problem, and thus we were able to simplify the conceptual problems associated with the constraints of the sequencing problem.

It is now necessary to obtain a mathematical expression for the objective function; that is, the total delay costs associated with a given sequence. First, the penalty cost associated with each job will be considered. In order to compute the penalty cost associated with job i, it is necessary to know the period in which job i is completed. Let T_i be the completion period of job i. Now, if we define C_i as the penalty cost associated with job i, we have

$$C_i = \begin{cases} p_i\ (T_i - g_i) & \text{if } T_i > g_i \\ 0 & \text{otherwise.} \end{cases} \qquad \text{Eq. 3.}$$

Now T_i is the sum of the processing times of all jobs done before job i plus the processing time of job i, d_i. Job j is done before job i if for any one position k, x_{jk} has a value 1 and x_{iq} has a value 0 for $q = 1,2,\ldots,k$. Thus if we define

$$u_{ik} = \sum_{q=1}^{k} x_{iq}, \qquad \text{Eq. 4.}$$

we can say that job j is done before job i if the indicator $x_{jk}(1-u_{ik})$ is 1 for any k. Note that the design of an indicator which specifies when job j is done before job i is a complex process. However, the mathematics involved in obtaining the design are straightforward. What was necessary was an ability to manipulate the variables as defined and to put together the variables in a form which resulted in the desired relationship. This is a design problem.

With the above information, an equation for the completion time of job i can be written as

$$T_i = d_i + \sum_{j=1}^{n} d_j \left\{ \sum_{k=1}^{n} x_{jk}(1 - u_{ik}) \right\}. \qquad \text{Eq. 5.}$$

In Equation 5, we have a product of unknowns which would yield quadratic terms in the objective function. These quadratic terms can be represented in a linear form by following a procedure developed by Watters [22].

If a quadratic term exists, say ab, where a and b are 0-1 variables then define f = ab, where f is a 0-1 variable. The truth table for f is given in Table IV.

Table IV

TRUTH TABLE FOR A QUADRATIC TERM

a	b	f
0	0	0
0	1	0
1	0	0
1	1	1

The procedure is to replace the quadratic term with f and to write f as a linear function of a and b. The inequalities for writing f in terms of a and b are

$$f \geq a + b - 1,$$

$$\text{and} \quad f \leq \tfrac{1}{2}(a + b).$$

The fact that f does indeed represent ab can be verified by exhaustion (recall that f is a 0-1 variable).

To use this procedure in the sequencing problem, replace $x_{jk}(1-u_{ik})$ by a newly defined 0-1 variable w_{ijk} in Equation 5,

$$T_i = d_i + \sum_{j=1}^{n} d_j \sum_{k=1}^{n} w_{ijk}, \qquad \text{Eq. 6.}$$

and impose the constraints

$$w_{ijk} \geq \left[x_{jk} + (1 - u_{ik})\right] - 1$$

and

$$w_{ijk} \leq 1/2\left[x_{jk} + (1 - u_{ik})\right].$$

Using Equation 4 for u_{ik} yields

$$w_{ijk} \geq x_{jk} - \sum_{q=1}^{k} x_{iq} \qquad \text{Eq. 7.}$$

and

$$w_{ijk} \leq \frac{1}{2}\left[x_{jk} + 1 - \sum_{q=1}^{k} x_{iq}\right]. \qquad \text{Eq. 8.}$$

Returning to the definition of the penalty cost for job i given in Equation 3, it is seen that the penalty cost can be either of two values depending upon the completion time of job i and the due date for job i. These conditions can be expressed in a single equation by defining a new 0-1 variable, say [illegible]$_i$, where

$$\delta_i = \begin{cases} 1 \text{ if job } i \text{ is late, i.e., } T_i > g_i \\ 0 \text{ otherwise.} \end{cases}$$

From this definition of δ_i, we can obtain the appropriate value of δ_i by requiring

$$\delta_i \geq \frac{T_i - g_i}{T}, \qquad \text{Eq. 9.}$$

and

$$\delta_i \leq 1 + \frac{T_i - g_i - 1}{T} \qquad \text{Eq. 10.}$$

$$\text{where } T = \sum_{i=1}^{n} d_i .$$

Equation 3 can now be rewritten as

$$C_i = p_i(T_i - g_i)\delta_i \qquad \text{Eq. 11.}$$

Substituting T_i into Equation 11 yields

$$C_i = p_i \left[d_i + \sum_{j=1}^{n} d_j \sum_{k=1}^{n} w_{ijk} - g_i \right] \delta_i$$

$$= p_i(d_i - g_i)\,\delta_i + p_i \sum_{j=1}^{n} d_j \sum_{k=1}^{n} \delta_i w_{ijk}$$

which again has a product of the unknowns included. To obtain a linear form, replace $\delta_i w_{ijk}$ with v_{ijk}, a 0-1 variable, where

$$v_{ijk} \geq \delta_i + w_{ijk} - 1$$

since if there is a choice for v_{ijk} it will be set to zero. This same observation can be made for w_{ijk}, and Equation 8 can be eliminated.

The above formulation is summarized below. The total penalty cost, Z, to be minimized is

$$Z = \sum_{i=1}^{n} \left[p_i (d_i - g_i)\, \delta_i + p_i \sum_{j=1}^{n} d_j \sum_{k=1}^{n} v_{ijk} \right] \qquad \text{Eq. 12.}$$

subject to

$$\sum_{k=1}^{n} x_{ik} = 1 \qquad i = 1,2,\ldots, n$$

$$\sum_{i=1}^{n} x_{ik} = 1 \qquad k = 1,2,\ldots, n$$

$$\delta_i \geq \frac{1}{T} \left[d_i + \sum_{j=1}^{n} d_j \sum_{k=1}^{n} w_{ijk} - g_i \right] \qquad i = 1,2,\ldots, n$$

$$\delta_i \leq 1 + \frac{1}{T} \left[d_i + \sum_{j=1}^{n} d_j \sum_{k=1}^{n} w_{ijk} - g_i - 1 \right] \qquad i = 1,2,\ldots, n$$

$$w_{ijk} \geq x_{jk} - \sum_{q=1}^{n} x_{iq} \qquad i, j, k = 1,2,\ldots, n \quad i \neq j,$$

$$v_{ijk} \geq \delta_i + w_{ijk} - 1 \qquad i, j, k = 1,2,\ldots, n \quad i \neq j$$

For the six-job example ($n = 6$) presented in Table I, there would be 36 values of x_{ik}, 6 values of δ_i, 180 values of w_{ijk} and 180 values of v_{ijk} for a total of 402 unknowns not including slack variables. There would be 384 constraints. Thus, the design does not appear to be efficient.

Back to the Drawing Board

Rather than attempt to polish the above design, it was decided to look at the problem from a different perspective [16]. Instead of looking at the sequence in which jobs are performed, the time during which the job is processed was examined. Again, other researchers have looked at the problem from this viewpoint [1, 10], and have defined the unknown variables as (1) the starting time of job i or (2) a 0-1 variable x_{it}, which is 1 in period t when job i is being processed and 0 otherwise. This latter definition has the appealing attribute of specifying which job is in process in any time period. However, it was recognized to be in the form of the difference of two step functions. It was thought that by defining two functions, one with a positive step at the start period and one with a positive step after the completion period, a savings in terms of the number of constraints could be obtained. Thus the following definitions were evolved:

t = period number where a period length is in the time units of the problem;

$$b_{it} = \begin{cases} 1 \text{ if job i is started prior to or at the beginning of period t} \\ 0 \text{ otherwise;} \end{cases}$$

$$x_{it} = \begin{cases} 1 \text{ if job i is completed prior to period t} \\ 0 \text{ otherwise;} \end{cases}$$

and T = time horizon and is equal to $\sum_{i=1}^{n} d_i$ for the one machine sequencing problem.

The above definitions yield a different discrete representation of the problem where every time period is counted as opposed to every sequencing position.

The constraints for this formulation involve the limiting of only one job being processed in any period and the maintaining of the start and completion time indicators. To restrict processing to only one job in a period, use is made of the fact that job i is processed in period t if and only if $(b_{it} - x_{it}) = 1$. Thus, this constraint for each t is

$$\sum_{i=1}^{n} (b_{it} - x_{it}) = 1, \qquad t = 1,2,\ldots T. \qquad \text{Eq. 13.}$$

To maintain the step functions, it is required that

$$b_{it} \le b_{i(t+1)} \qquad t = 1,2,\ldots, T-1; \text{ all } i \qquad \text{Eq. 14.}$$

and

$$x_{it} \le x_{i(t+1)} \qquad t = 1,2,\ldots, T-1; \text{ all } i. \qquad \text{Eq. 15.}$$

Since job continuity is required, the job is completed d_i time units after it is started and

$$b_{it} = x_{i(t+d_i)} \qquad \text{all } i, t \qquad \text{Eq. 16.}$$

where

$$x_{it} = 1 \qquad \text{for } t > T.$$

The ability to use d_i as a subscript is predicated on the assumption of a deterministic processing time. This translation operation and removal of one of the step functions is obvious now, but was not during the design stages. The constraints can be rewritten as

$$x_{it} = 0 \qquad t \le d_i; \quad \text{all } i \qquad \text{Eq. 17.}$$

$$x_{it} = 1 \qquad t = T+1, \ldots, T+d_i; \quad \text{all } i \qquad \text{Eq. 18.}$$

$$x_{it} \le x_{i(t+1)} \qquad t = d_i + 1, \ldots, T; \quad \text{all } i \qquad \text{Eq. 19.}$$

and

$$\sum_{i=1}^{n} (x_{i(t+d_i)} - x_{it}) = 1 \qquad t = 1,2,\ldots, T. \qquad \text{Eq. 20.}$$

Equation 17 states that job i cannot be completed prior to a period which is not larger than its duration. Actually, Equation 17 is definitional and reduces the number of unknowns of the problem. Equation 18 states that all jobs are done within the time horizon. It is also definitional and is inserted directly into Equations 19 and 20. Equation 20 is rewritten from Equations 13 and 16. With this redesign, Equation 14 is eliminated. (This corresponds to a reduction of n(T-1) variables and constraints.)

Turning now to the objective function, job i is late in period t ($t > g_i$) if x_{it} is 0. Thus the lateness cost associated with job i, C_i, is

$$C_i = p_i \sum_{t=g_i+1}^{T} (1 - x_{it}).$$

The objective can be stated as the minimization of the function Z where

$$Z = \sum_{i=1}^{n} p_i \sum_{t=g_i+1}^{T} (1 - x_{it}) \qquad \text{Eq. 21.}$$

or, equivalently, the maximization of

$$z = \sum_{i=1}^{n} p_i \sum_{t=g_i+1}^{T} x_{it}. \qquad \text{Eq. 22.}$$

The number of unknowns for this formulation is (n-1)T and the number of constraints is nT. For the six-job example problem, T = 26 and n = 6, and there are 130 unknowns and 156 constraints. This is a significant reduction from the previous formulation. In Table V, the values of the variables are shown for the optimal solution.

Icing on the Cake

In examining the formulation presented in the above section, it is seen that Equation 19 represents (n-1)T of the nT constraints. Thus, Equation 19 deserves further scrutiny [17]. Equation 19 is required to maintain the step function nature of the 0-1 variable x_{it} over the time horizon. All that is really desired is a knowledge of the period at the end of which each job is completed. Based on the first design, it appeared plausible to consider a definition for the unknown variable which was 1 for the period that the job is completed and 0 otherwise. This design of the unknown combines the previous definitions of the unknowns. That is, it involves the completion of the job in the definition, it involves the division of the time horizon into periods, and it involves the assignment of job completion periods to specific time periods. The new variable will be y_{it}, and is defined as

$$y_{it} = \begin{cases} 1 \text{ if job } i \text{ is completed at the end of period } t \\ 0 \text{ otherwise} \end{cases}$$

TABLE V

OPTIMAL SOLUTION FOR SIX-JOB, ONE-MACHINE EXAMPLE

x_{it}

i \ t	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	C_i
1	-	- |	-	-	-	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	+	+	+	+	+			35.0
2	-	-	-	- |	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	+	+	+	+				0.0
3	-	-	-	0	0	0	0	0 |	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	+	+	+					12.0
4	-	-	-	-	-	0	0	0	0	0	0	0 |	0	0	0	0	0	0	0	0	0	0	0	0	0	0	+	+	+	+	+			14.0
5	-	-	0	0	0	0	0	0	0	0	0	1	1 |	1	1	1	1	1	1	1	1	1	1	1	1	1	+	+						0.0
6	-	-	-	-	-	-	-	0	0	0	0	0	0	0	0	0	0 |	0	0	0	0	1	1	1	1	1	+	+	+	+	+	+	+	8.0

TOTAL PENALTY COST 69.0

NOTE: - indicates value defined as zero
+ indicates value defined as one
| indicates the due date period

Since job i cannot be completed prior to $t = d_i$, we have

$$y_{it} \equiv 0 \qquad t < d_i; \qquad \text{all } i. \qquad \text{Eq. 23.}$$

The relationship between the x_{it} variable defined in the previous section and y_{it} is

$$x_{it} = \sum_{q=d_i}^{t-1} y_{iq} \qquad \text{Eq. 24.}$$

Consider now the conversion of Equations 18-20 to constraints involving y_{it}. Equation 18 specified that all jobs were completed in the time horizon. It is now required that each job be assigned one and only one completion period. Thus

$$\sum_{t=d_i}^{T} y_{it} = 1 \qquad \text{all } i \qquad \text{Eq. 25.}$$

and

$$y_{it} \equiv 0 \qquad t > T. \qquad \text{Eq. 26.}$$

Equation 26 is definitional and states that completion of the job cannot occur after period T. Since the step function concept is not being used, Equation 19 requires no equivalent in the new formulation. The equivalent form of Equation 20 can be obtained by substituting Equation 24 into Equation 20 to obtain

$$\sum_{i=1}^{n} \left[\sum_{q=d_i}^{t+d_i-1} y_{iq} - \sum_{q=d_i}^{t-1} y_{iq} \right] = 1$$

or

$$\sum_{i=1}^{n} \sum_{q=t}^{t+d_i-1} y_{iq} = 1 \qquad t = 1, 2, \ldots, T. \qquad \text{Eq. 27.}$$

Similarly, Equation 22 Becomes

$$z = \sum_{i=1}^{n} p_i \sum_{t=g_i+1}^{T} \sum_{q=d_i}^{t-1} y_{iq}. \qquad \text{Eq. 28.}$$

For this reformulation, the number of unknowns has not been reduced, i.e., there are (n-1)T values of y_{it} to be determined. However, the number of explicit constraints as represented by Equations 25 and 27 is only n + T. (Note that Equations 24 and 26 are definitional.) Thus, for the six-job example, the problem has been reduced to 130 nonslack variables and 32 constraints. Perhaps more significant is the observation that the number of constraints only increases as the sum of n and T. (Actually the n equations represented by Equation 25 can also be removed by solving for y_{iT} and substituting into Equation 27.)

Generalizations

The model discussed in this paper is actually a portion of a model developed to study the scheduling of projects consisting of a network of jobs under the conditions of limited resource availability [16, 17]. Jobs were permitted to require multiple resources also. Equations were developed to represent the following objectives:

1. Minimize the sum of the throughput time (time in the shop) for all projects;
2. Minimize the time by which all projects are completed (minimize makespan); and

3. Minimize the sum of penalty costs (a generalization of the p_i in this paper to allow a different penalty in each period, viz., p_{it}.)

The following constraints were also modeled:

1. Limited resources;
2. Precedence relations between jobs;
3. Job splitting possibilities (interrupts);
4. Project and job due dates;
5. Substitution of resources to perform the jobs; and
6. Concurrent and nonconcurrent performance of jobs.

Conclusion

The formulation of problems in mathematical terms is a design activity which requires an intimate knowledge of the problem being studied and an ability to evolve novel approaches to the design.

BIBLIOGRAPHY

1. Bowman, E. H., "The Schedule Sequencing Problem," Operations Research, Vol. 7, 1959, pp. 621-624.

2. Davis, E. W., "Resource Allocation in Project Network Models--A Survey," Journal of Industrial Engineering, Vol. 17, April 1966, pp. 177-188.

3. Elmaghraby, S. E., "A Note on the One Machine Sequencing Problem With Delay Costs," to be published.

4. Fendley, L. G., "The Development of a Complete Multi-Project Scheduling System Using a Forecasting and Sequencing Technique," Ph.D. Dissertation, Arizona State University, 1967.

5. Gere, W., "Heuristics in Job-Shop Scheduling," Management Science, Vol. 13, No. 3, November 1966, pp. 167-190.

6. Hadley, G., Nonlinear and Dynamic Programming, Reading, Massachusetts, Addison-Wesley Publishing Co., Inc., 1964.

7. Held, M. and R. Karp, "A Dynamic Programming Approach to Sequencing Problems," Journal of SIAM, Vol. 10, No. 1, 1962.

8. Lambourn, S., "RAMPS--A New Tool in Planning and Control," The Computer Journal, Vol. 5, 1963, pp. 300-304.

9. Lawler, E. L., "On Scheduling Problems with Deferral Costs," The Computer Journal, Vol. 5, 1963, pp. 300-304.

10. Manne, A. S., "On the Job-Shop Scheduling Problem," Operations Research, Vol. 8, 1960, pp. 219-223.

11. McNaughton, Robert, "Scheduling with Deadlines and Loss Functions," Management Science, Vol. 6, No. 1, October 1959.

12. Mize, J. H., "A Heuristic Scheduling Model for Multi-Project Organizations," Ph.D. Thesis, Purdue University, 1964.

13. Moder, J. J., and C. R. Phillips, Project Management With CPM and PERT, New York, Reinhold Publishing Company, 1964.

14. Montagne, E. R., Jr., "An Investigation of Sequencing with Time Delay Costs," Unpublished Master's Report, Arizona State University, January 1968.

15. Muth, J. F., and G. L. Thompson, (Eds.), Industrial Scheduling, Englewood Cliffs, N. J., Prentice-Hall, Inc., 1963.

16. Pritsker, A. A.B., and L. J. Watters, A Zero-One Programming Approach to Scheduling with Limited Resources, The RAND Corporation, RM-5561-PR, January 1968.

17. Pritsker, A.A.B., L. J. Watters, and P. M. Wolfe, "Multiproject Scheduling with Limited Resources; A 0-1 Programming Approach," A RAND Corporation Paper, February 1968.

18. Root, J. G., "Scheduling with Deadlines and Loss Functions," Management Science, Vol. 11, No. 3, January 1965.

19. Sisson, R. L., "Sequencing Theory" in Progress in Operations Research, Vol. I, R. L. Ackoff (Ed.), New York, John Wiley and Sons, Inc., 1961.

20. Smith, R. D., and R. A. Dudek, "A General Algorithm for Solution of the n-Job, M-Machine Sequencing Problem of the Flow Shop," Operations Research, Vol. 15, No. 1, January-February 1967, pp. 71-82.

21. Wagner, H., "An Integer Linear Programming Model for Machine Scheduling," Naval Research Logistics Quarterly, Vol. 6, 1959, pp. 131-140.

22. Watters, L. J., "Reduction of Integer Polynomial Programming Problems to Zero-One Linear Programming Problems," Operations Research, Vol. 15, No. 6, November-December 1967, pp. 1171-1174.

23. Wiest, J. D., "Some Properties of Schedules for Large Projects with Limited Resources," Operations Research, Vol. 12, May-June 1964, pp. 395-418.

24. Wiest, J. D., "A Heuristic Model for Scheduling Large Projects with Limited Resources," Management Science, Vol. 13, No. 6, February 1967, pp. B-359-377.