Division of Research
Graduate School of Business Administration
The University of Michigan

March 1982

NONLINEAR MIP FORMULATIONS OF PRODUCTION PLANNING PROBLEMS IN FLEXIBLE MANUFACTURING SYSTEMS

Working Paper No. 293

Kathryn E. Stecke

The University of Michigan

FOR DISCUSSION PURPOSES ONLY

ABSTRACT

A flexible manufacturing system (FMS) is an integrated, computer-controlled complex of automated material handling devices and numerically controlled machine tools that can simultaneously process medium-sized volumes of a variety of part types. FMSs are becoming an attractive substitute for the conventional means of batch manufacturing, especially in the metal-cutting industry. This new production technology has been designed to attain the efficiency of well-balanced, machine-paced transfer lines, while utilizing the flexibility that job shops have to simultaneously machine multiple part types. Some properties and constraints of these systems are similar to those of flow and job shops, while others are different. This technology creates the need to develop new and appropriate planning and control procedures that take advantage of the system's capabilities for higher production rates.

This paper defines a set of five production planning problems that must be solved for efficient use of an FMS, and addresses specifically the grouping and loading problems. These two problems are first formulated in detail as nonlinear 0-1 mixed integer programs. In an effort to develop solution methodologies for these two planning problems, several linearization methods are examined and applied to data from an existing FMS. To decrease computational time, the constraint size of the linearized integer problems is reduced according to various methods. Several problems are solved using the linearization that results in the fewest additional constraints and/or variables. The problem characteristics that determine which linearization to use, and the application of the linearized models in the solution of actual planning problems, are also discussed.

1. Introduction

Approximately 50 percent of U.S. annual expenditures on manufacturing is in the metal-working industry, and two-thirds of metal-working is metal cutting. In addition, approximately 75 percent of the dollar volume of metal-worked products is manufactured in batches of less than 50 parts (Cook [1975]). Of late, the industry has become concerned with the very low productivity of these mid-volume systems.

Until recently, and especially relative to the development of methods for the mass production of a single part type, little attention had been given to batch manufacturing. The growth of the metal-working industry spawned technological improvements over time, such as the use of numerically controlled (NC) machines, which spurred research into the development of efficient means for small-batch production.

One result was the development of flexible manufacturing systems (FMSs). By 1976, four FMSs were in operation in the U.S. and several in Europe. Japan leads in terms of the number of FMSs, but their systems are generally simpler than those in the U.S. The number of new systems in the U.S. is expected to grow rapidly. In fact, it has been estimated that about 5,000 such systems will be in existence by the year 2000 (Barash [1978]). A description of several newer FMSs can be found in Barash [1982], and additional information is contained in Stecke and Solberg [1981].

The aim of an FMS is to achieve the efficiency of automated mass production, while utilizing the flexibility of a manual job shop to simultaneously machine several part types. An FMS is an automated batch manufacturing system consisting of NC machines, linked by automated material handling, that perform the operations required to manufacture parts. Each operation requires one or more cutting tools. The tools for all operations that can be performed by a machine are stored in its limited capacity tool magazine. Each machine has an automatic tool interchanging device that can interchange two cutting tools in seconds. This rapid interchange capability allows several consecutive operations to be machined with virtually no set-up time between operations. One or more computers control most activity, such as the machining operations, part movements, and tool interchanges. The computer cannot route a particular part to a machine unless all tools required for the next operation have been previously placed in the magazine. This last requirement indicates the need for planning prior to production.

Because the concepts and technology of automated batch manufacturing are still in their infancy, problems have been encountered. The need for precise planning is noted by the initial FMS users (Berdine [1978]) and manufacturers (Dirksen [1977]).

Managing production for an FMS is more difficult than for production lines and job shops because (1) each machine is quite versatile and capable of performing many different operations, (2) the system can machine several part types simultaneously, and (3) each part may have alternative routes through the system. These additional capabilities and planning options increase both the number of decision variables and the constraints associated with setting up an FMS.

To best utilize an FMS's capabilities, a careful system set-up is required prior to production. Set-up decisions for batch manufacturing are made frequently to meet new or altered production requirements, for example, whenever production requirements for a part type are met. This contrasts with a mass production system where set-up is part of the design process and set-up changes are few.

The decision variables of the set-up problem of an FMS are: part types to be produced next, relative numbers of parts of each type to be machined simultaneously, number of pallets and fixtures of each fixture type to be reserved for each part type, and allocation of operations (and tools) to machines. The objective of a set-up is system-dependent, but commonly it is to maximize expected production.

The initial systems were managed by means of conventional loading and scheduling methods, such as assigning each operation to only one machine and attempting to balance the assigned workload per machine. It seemed to us that perhaps new loading and control strategies could be developed to take advantage of the machine capabilities and system flexibility. As a result, alternative loading and scheduling strategies were defined (Stecke and Solberg [1982b], which proved better than the conventional methods when applied to a detailed simulation of an existing FMS. The results were surprising because the resulting workloads were highly unbalanced.

Since the set-up problem in its general form is intractable, the following framework is suggested to help a manager in setting up his FMS for efficient production. Five production planning problems are defined here, the solutions to which comprise a system set-up. The problems can be solved sequentially or, alternatively, candidate solutions to the problems can be generated iteratively until a suitable final solution is found. In addition, surrogate objectives, rather than a direct attempt to maximize production, are used for each problem. The problems are:

1. Part type selection problem:
   From a set of part types that have production requirements, determine a subset for immediate and simultaneous processing.

2. Machine grouping problem:
   Partition the machines into machine groups in such a way that each machine in a particular group is able to perform the same set of operations.

3. Production ratio problem:
   Determine the relative ratios at which the part types selected in problem (1) will be produced.

4. Resource allocation problem:
   Allocate the limited number of pallets and fixtures of each fixture type among the selected part types.

5. Loading problem:
   Allocate the operations and required tools of the selected part types among the machine groups subject to technological and capacity constraints of the FMS.

This paper addresses the problems of <u>grouping</u> and <u>loading</u>. For additional information concerning all of these problems, see Stecke [1981].

The plan of the paper is as follows. The machine grouping (2) and loading (5) problems are formally defined in §2 as 0-1 nonlinear mixed integer programs (MIPs). In addition, different methods of solving nonlinear MIPs are surveyed. §3 presents several linearization methods. In §4, these are applied to data from an existing FMS. Numerical solutions are obtained by using the linearization that results in the fewest additional constraints and/or variables. In addition, conclusions are drawn concerning conditions under which each linearization is best. §4 also includes a discussion of computational experience, which shows the advantages of considering the nonlinear terms. §5 presents a summary and conclusions.

2. Mathematical Programming Formulations

After defining required notation, this section begins by developing the constraint formulations necessary for the grouping and loading problems. The grouping problem is then defined and formulated. Finally, several loading objectives are developed.

Parameters and Variables

The subscripts, input parameters, and decision variables are given in Table 1. Several subscript conventions and parameters require further explanation. In the formulations, several of the parameters given in Table 1 might apply to either machines (j) or machine groups (ℓ). In those formulations for which there is only one machine in each group, the machine subscript j is used.

The differences between machine types and machine groups should be clarified. All machines of the same machine type are physically identical (i.e., they have the same axes of motion, dimensions, horsepower, capabilities). Each set of machine types, m_n, might be partitioned into several machine groups. Machines that are identically tooled comprise a group and are said to be pooled. Hence, each machine in a particular group will be able to perform the same operations.

TABLE 1

Notation

Subscripts:

operation	$i = 1,\ldots, b$
machine	$j = 1,\ldots, m$
machine group	$\ell = 1,\ldots, \underline{M}$
machine type	$n = 1,\ldots, \overline{m}$
set of operations	$k = 1,\ldots, K$

Parameters (Input):

$p_{i\ell}$ = processing time of operation i on one of the machines in machine group ℓ

q_i = maximum number of times that operation i can be assigned

d_i = number of slots required in a tool magazine by operation i

t_ℓ = capacity of the tool magazine for each machine in group ℓ

w_{ik} = number of slots saved as a result of having common tools when operations i and k are assigned to the same machine
= count of the number of spaces (slots) occupied by the tools contained in the intersection of the sets of tools required by operations i and k

B_k = index set of sets of operations

$\overline{B}$ = index subset of B_k such that $|\overline{B}|$, the cardinality of $\overline{B}$, = p, $p = 2,\ldots, b$

w_{B_k} = number of slots saved when the operations in B_K are assigned to the same machine

P = index set of compatible part types that are to be produced simultaneously on the system of machines

a_i = production ratio (relative to the remaining part types in P-{i}) at which part type i will be produced

m_n = $\{j \mid \text{machine } j \text{ is of machine type } n\}$

Decision Variables (Output):

M_ℓ = $\{j \mid \text{machine } j \text{ is in machine group } \ell\}$

$$x_{i\ell} = \begin{cases} 1, & \text{if operation } i \text{ is assigned to each machine in group } \ell; \\ 0, & \text{otherwise.} \end{cases}$$

Constraint Formulations

The constraints of the grouping and loading problems are as follows.

First, each operation must be assigned to at least one machine of the machine type required by the operation. In addition, there is a limitation on the number of duplicate assignments allowed:

$$1 \leq \sum_{\ell=1}^{M} x_{i\ell} \leq q_i, \quad i = 1,\ldots, b. \tag{1}$$

It is understood that $x_{ij}(x_{i\ell}) = 0$ if operation i cannot be performed by the machine type corresponding to machine (group) j or ℓ.

Second, the tool magazine capacity constraint, which relates the number of tool slots required by the operations assigned to a machine (group) to the total number of slots contained in the machine (group)'s tool magazine, in its simplest form, is

$$\sum_{i=1}^{b} d_i x_{i\ell} \leq t_\ell, \quad \ell = 1,\ldots, m.$$

Since only one tool can be used at a time, however, it is unnecessary to assign any tool more than once to the same machine. Also, the actual number of slots used depends on the physical placement of the tools in the tool magazine. In the example shown in Figure 1, two three-slot tools placed side by side require only five slots rather than six. Another complicating consideration is that since larger tools are heavier, tool magazines must be weight balanced. In addition, several operations may require some of the same tools. Space in the tool magazine can be saved by eliminating tool duplication and considering overlap and weight balancing. These savings are

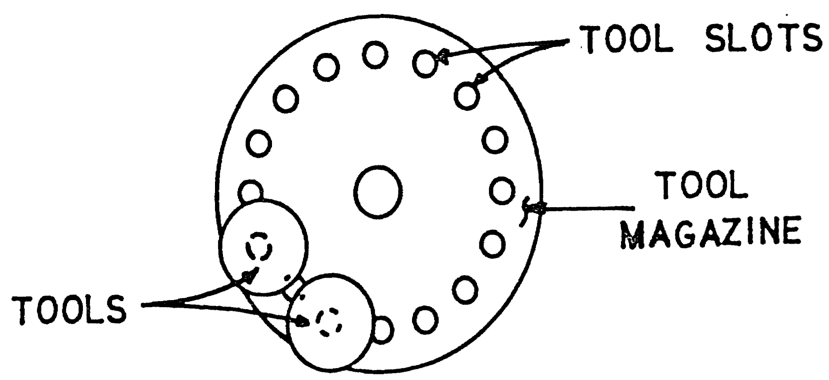

Figure 1. Tool Magazine

measured by w_{B_k}. The tool magazine capacity constraint then becomes:

$$\sum_{i=1}^{b} d_i x_{i\ell} - \sum_{i_1=1}^{b-1} \sum_{i_2=i_1+1}^{b} w_{i_1 i_2} x_{i_1\ell} x_{i_2\ell} +$$

$$+ \sum_{i_1=1}^{b-2} \sum_{i_2=i_1+1}^{b-1} \sum_{i_3=i_2+1}^{b} w_{i_1 i_2 i_3} x_{i_1\ell} x_{i_2\ell} x_{i_3\ell} + \ldots$$

$$+ (-1)^{p+1} \sum_{i_1=1}^{b-p+1} \sum_{i_2=i_1+1}^{b-p+2} \cdots \sum_{i_p=i_{p-1}+1}^{b} w_{i_1 i_2 \ldots i_p} x_{i_1\ell} x_{i_2\ell} \ldots x_{i_p\ell} \leq t_\ell,$$

or, in more compact form,

$$\sum_{i=1}^{b} d_i x_{i\ell} + \sum_{p=2}^{b} (-1)^{p+1} \sum_{\substack{\forall \overline{B} \subseteq B_k \\ \ni |\overline{B}|=p}} w_{\overline{B}} \prod_{i_k \in \overline{B}} x_{i_k\ell} \leq t_\ell, \quad \ell = 1, \ldots, M. \tag{2}$$

Finally, there is the integrality constraint:

$$x_{i\ell} = 0 \text{ or } 1, \text{ for all } i, \ell. \tag{3}$$

Machine Grouping Formulation

Pooling (see Kleinrock [1976], Stecke and Solberg [1981]) increases system performance by decreasing the probability that a part will be blocked by having no machine available for the next operation. Having more than one machine in a group is one way to allow alternate routes for some part types.

Stecke and Solberg [1982a] consider the best partitions of m items (servers, machines) into M (machine) groups to maximize expected production using a closed queueing network model. In particular, the results include:

i. Fewer groups are better; i.e., pool as much as possible.

ii. The maximum expected production is obtained from systems with the most unequal sized groups. More generally, all possible partitions are ordered according to expected production.

These results are summarized in the Appendix and are used here shortly.

However, the grouping problem considered in this paper is more complex than that in Stecke and Solberg [1982a], because additional constraints on tool requirements and tool magazine capacity that use actual operation times have been imposed.

Maximum pooling of all machines of the same type into one group may not be possible because of some technological constraints such as tool magazine capacity. The approach we take to maximize pooling is as follows:

1. Set M equal to the minimum number of machines (or machine groups) required to perform all operations of the part types in P.
2. Use the optimal pooling for M groups from Stecke and Solberg [1982a] (provided in the Appendix).

Since all possible partitions of machines are ordered, the solution to the more detailed model is then the best feasible partition.

Step 1, which determines M, is now described. The problem is to find $\{x_{ij} | i=1,\ldots,b,\ j=1,\ldots,m\}$ to

$$\text{maximize} \quad \sum_{j=1}^{m} \gamma^{j} s\ell_j$$

subject to

$$s\ell_j = t_j - \Big(\sum_{i=1}^{b} d_i x_{ij} + \sum_{p=2}^{b} (-1)^{p+1} \sum_{\substack{\forall \bar{B} \subseteq B_k \\ \ni |\bar{B}|=p}} w_{\bar{B}} \prod_{i_k \in \bar{B}} x_{i_k j}\Big),$$

and equations (1), (3), and $q_i = 1$,

where $\gamma = \sum_{i=1}^{b} d_i$. ($\gamma$ is merely a large number, so that γ^j increasingly weights the slack in the tool magazine capacity constraint of machine j.)

Let m_{on} be the maximum number of machines of machine type n required to perform the operations of the part types in P, if common tooling is not taken into consideration (n=1,...,m). The m_{on} are obtained by adding the number of tool slots required of machine type n, dividing by the capacity of each machine's tool magazine, t_n, and rounding to the next highest integer. Then an upper bound on the total number of required machines is m_o machines, where

$$m_o = \sum_{n=1}^{\bar{m}} m_{on}.$$

The objective function formulation allows for the values of the slack ($s\ell_j$) in the $|m_n|$ machines to monotonically increase. For every machine type, the initial machines will be filled first; if there is insufficient tool slot capacity and another machine is required, machine i will tend to be filled before an operation is assigned to machine j, for $i < j$. The result is the minimum number of machines of each type that are needed to perform the required operations.

An example will demonstrate the procedure. Consider a 15-machine system of four machine types with $m_{o1} = 4$, $m_{o2} = 3$, $m_{o3} = 4$, and $m_{o4} = 3$. Then 14 machines are required if overlap is not considered. The machines, j, and their machine types, n, are as follows:

n:	1	2	3	4
machine j:	(1 2 3 4)	(5 6 7)	(8 9 10 11 12)	(13 14 15)

Suppose that the solution to Step 1 was that M = 10, and that three machines of each of the first three types were required, and only one of the fourth type. Then the optimal pooling into machine groups according to Step 2 is:

machine j: (1 2) (3) (4) (5) (6) (7) (8 9 10) (11) (12) (13 14 15)

Notice that all machines of the fourth type could be pooled, none of the second type could be, and there are 10 machine groups.

Loading Formulations

A usual loading procedure for both conventional systems and FMSs attempts to balance the assigned workload on each machine; the aim is to equalize the total weighted processing time, or workload, of the operations assigned to each machine. The processing time of each operation is weighted by the production ratio (a_i) of the corresponding part type i, as calculated in the fourth production planning problem. In addition, each operation is often assigned to only one machine. The consequence is that each part type has a fixed route through the shop.

However, the flexibility and capabilities of an FMS indicate that perhaps new planning and control procedures should be developed for FMSs, which would also perhaps be applicable to other types of manufacturing systems. In a previous study (Stecke [1977]), alternative loading objectives were defined. Application of any of several objectives can result in better system performance than attempting to balance assigned workloads.

Stecke [1981] has shown that the practice of balancing is too restrictive for most FMSs, since the inherent flexibility can often be utilized for better system performance. Several alternative loading objectives are listed in Table 2. The decision concerning which to apply is problem-dependent. Each may be best under certain circumstances. In some situations, some of the objectives are contradictory, while in others, several objectives may be equally applicable.

TABLE 2

Loading Objectives

1. Balance the assigned machine processing times.

2. Minimize the number of movements from machine to machine, or equivalently, maximize the number of consecutive operations on each machine.

3. Balance the workload per machine for a system of groups of pooled machines of equal sizes.

4. Unbalance the workload per machine for a system of groups of pooled machines of unequal sizes.

5. Fill the tool magazines as densely as possible.

6. Maximize the sum of operation priorities.

Consider the first objective function, balancing the assigned machine processing times. Let r_j be the relative workload assigned to machine j:

$$r_j = \sum_{i=1}^{b} a_i p_{ij} x_{ij}, \quad j = 1,\ldots, m. \tag{5}$$

Then r_j is also a measure of the relative utilization of machine j. If all of the r_j are equal, the system is perfectly balanced. This is usually not possible because of the discrete values of processing times. The following formulations optimally balance the workloads assigned to machines as much as possible. Each objective function is a measure of system imbalance.

The problem is to find $\{x_{ij} | \; i=1,\ldots, b, \; j=1,\ldots, m\}$ to minimize $h_i(x)$, where $h_i(x)$ is one of the following four:

1. $\underset{\substack{j=1,\ldots,m-1 \\ h=1,\ldots,m;}}{\text{maximum}} |r_j - r_h|$

2. $\sum_{j=1}^{m-1} \sum_{h=j+1}^{m} |r_j - r_h|^{\gamma}, \; \gamma > 0$

3. $\sum_{j=1}^{m-1} \sum_{h=j+1}^{m} (r_j - r_h)^2$

4. $\beta - \alpha$,
   subject to $0 \leq \alpha \leq r_j \leq \beta, \qquad j = 1,\ldots, m.$

The constraints are: (1), (2), (3), (5), and $q_i = 1$.

The second objective, minimizing the number of movements, is quite different from the first. It is relevant, for example, when transportation time or distance from machine to machine is large relative to average operation time. There are manufacturing systems for which minimizing movements from machine to machine is preferable, even at the expense of balancing (Stecke and Solberg [1982b]). It can be more advantageous for a part type to remain on a machine for several consecutive operations rather than to move for the sake of balancing. Furthermore, when several consecutive operations require the same machine type, time may be saved by processing all of them on the same machine, if this is technologically possible. Both travel time (from machine to machine) and waiting time (for a subsequent, possibly busy machine to become idle) may be saved.

The first two objectives given in Table 2 are often incompatible. When operations are being allocated in large sets, the potential for balance decreases: if the operation times to allocate are smaller, a better balance is likely.

We now formulate the second loading objective. Notice that if i and i+1 represent consecutive operations, then

$$x_{ij} - x_{i+1,j} = \begin{cases} 0, & \text{if operations } i \text{ and } i+1 \text{ are on the same machine } j; \\ \pm 1, & \text{if operations } i \text{ and } i+1 \text{ are assigned to different machines.} \end{cases}$$

If N is defined as twice the number of excess movements, then

$$N = \sum_{i=1}^{b-1} \sum_{j=1}^{m} |x_{ij} - x_{i+1,j}| = \sum_{i=1}^{b-1} \sum_{j=1}^{m} (x_{ij} - x_{i+1,j})^2.$$

Some of the differences $(x_{ij} - x_{i+1,j})$ need not be included in the calculation of N. For example, for some machine j, operation i may require a machine type other than that of machine j; in this case, $x_{ij} = 0$. In particular, if $x_{ij} - x_{i+1,j} = x_{ij}$ or $-x_{i+1,j}$, then this term may be excluded from N. Inclusion is not incorrect, merely unnecessary and inefficient. The second objective, then, is to

$$\text{minimize} \sum_{i=1}^{b-1} \sum_{j=1}^{m} (x_{ij} - x_{i+1,j})^2 \text{ (or)}$$

$$= \sum_{i=1}^{b-1} \sum_{j=1}^{m} |x_{ij} - x_{i+1,j}|,$$

subject to (1), (2), (3), and $q_i = 1$.

The advantages from utilizing flexibility by allowing pooling and hence alternative part routes provide motivation for the remaining loading objectives. The third (and fourth) objectives are to (un)balance the workload per machine for a system of groups of pooled machines. The applicability of either depends on the configuration of the manufacturing system, or how the machines are grouped (see the Appendix). For the third objective, the problem is to minimize $h_i(x)$, where $h_i(x)$ is one of the following four:

1. $$\underset{\substack{\ell=1,\ldots,M-1 \\ k=\ell+1,\ldots,M}}{\text{maximum}} \left| \frac{r_\ell}{s_\ell} - \frac{r_k}{s_k} \right|$$

2. $$\sum_{\ell=1}^{M-1} \sum_{k=\ell+1}^{M} \left| \frac{r_\ell}{s_\ell} - \frac{r_k}{s_k} \right|^\gamma, \qquad \gamma > 0$$

3. $$\sum_{\ell=1}^{M-1} \sum_{k=\ell+1}^{M} \left(\frac{r_\ell}{s_\ell} - \frac{r_k}{s_k}\right)^2$$

4. $\beta - \alpha$,

   subject to $0 \leq \alpha \leq r_\ell/s_\ell \leq \beta$, $\qquad \ell = 1,\ldots, M$,

subject to (1), (2), (3), (5), and $q_i = 1$.

Notice that r_ℓ/s_ℓ is the workload per machine in machine group ℓ.

The fourth problem is to minimize $g_i(x)$, where $g_i(x)$ is one of the following four:

1. $\underset{\ell=1,\ldots,M}{\text{maximum}} \; |r_\ell - X^*_\ell|$
2. $\underset{\ell=1,\ldots,M}{\text{maximum}} \; |r_\ell - X^*_\ell|^\gamma, \quad \gamma > 0$
3. $\sum_{\ell=1}^{M} (r_\ell - X^*_\ell)^2$
4. $\beta - \alpha$,
   subject to $0 \leq \alpha \leq r_\ell - X^*_\ell \leq \beta, \qquad \ell=1,\ldots, M,$

subject to (1), (2), (3), (5), and $q_i = 1$, and where X^*_ℓ is the theoretical optimal workload that should be assigned to machine group ℓ to maximize expected production (see the Appendix).

The rationale for the fifth objective given in Table 2 is that when tool magazines are filled, perhaps several operation assignments may be duplicated to produce alternative part routes, which should increase machine utilization and production, and decrease waiting time. No single tool should be assigned to any particular machine more than once. In addition, the maximum number of times that an operation could be duplicated can be specified. One formulation of this objective minimizes slack in the capacity constraints for all machines. Then the problem is to

$$\text{minimize} \sum_{j=1}^{m} s\ell_j$$

subject to (1), (3), $q_i \geq 1$, and

$$s\ell_j = t_j - \Big(\sum_{i=1}^{b} d_i x_{ij} + \sum_{p=2}^{b} (-1)^{p+1} \sum_{\substack{\forall \overline{B} \subseteq B_k \\ \ni |\overline{B}|=p}} w_{\overline{B}} \prod_{i_k \in \overline{B}} x_{i_k j} \Big).$$

The aim of the sixth objective given in Table 2 is similar to the aim of the fifth: to duplicate assignments of some operations. Operation assignments should not be duplicated arbitrarily. Some operations, such as bottleneck operations, are more critical than others. In such cases, weights could be assigned to prejudice operation assignments. If w_i is the weight assigned to operation i, then the problem is to

$$\text{maximize} \sum_{i=1}^{b} \sum_{j=1}^{m} w_i x_{ij}$$

subject to (1), (2), (3), and $q_i \geq 1$.

Almost all of the objective functions and some of the constraints are nonlinear. A variety of approaches for solving nonlinear MIPs is available. They can be solved directly (Cooper [1981], Ginsburgh and Van Peetersen [1969], Hammer [1969], Hansen [1971, 1979], Marsten and Morin [1978], Morin [1978], Lawler and Bell [1967]). Alternatively, piecewise linear approximations (Dantzig [1962], Hu [1969], Watters [1967]) or heuristic algorithms (Stecke and Solberg [1981]) can be used. Another approximate approach is to ignore some tooling considerations that result in the nonlinear terms in the capacity constraints. Ignoring these factors results in feasible, but worse, solutions. An exact approach is to linearize the nonlinear terms (Balas [1964], Glover and Woolsey [1973, 1974], Glover [1975]). The resultant linearized 0-1 problems could either be solved directly or relaxed. The applicability of a direct nonlinear approach versus a transformed linear approach is problem dependent (Taha [1970]). The direct approach is more difficult, while linearization results in much larger problems. Finally, the linearized integer problems can be solved by means of fast approximation algorithms (Hochbaum [1980]).

In the following section, we present several methods to linearize the nonlinear terms. Combinations of these methods are applied to data in §4.

3. Linearization of the Product Terms

Fortunately, the nonlinear terms in the formulations are products of 0-1 integer variables. Several methods can be used to linearize the terms. The standard procedure is to replace each cross-product term with a new variable. Additional constraints are required to insure that the new variables take on the correct values.

In this section we survey five linearization methods, which differ in both the numbers of additional variables (either integer or continuous) and constraints generated. The difficulty of integer problems depends primarily on the number of integer, rather than continuous, variables. Additional details concerning the linearizations of the formulations in section 2 (in particular, the generated variables and constraints) can be found in Stecke [1981].

The first method was developed by Balas [1964]. Each product term $\prod_{i_k \in \bar{S}} x_{i_k j}$ is replaced by a new variable $x_{\bar{S}}$

$$\sum_{i_k \in \bar{S}} x_{i_k j} - x_{\bar{S}} \leq p - 1, \qquad p = |S| \tag{6}$$

$$\sum_{i_k \in \bar{S}} x_{i_k j} + p\, x_{\bar{S}} \leq 0, \qquad j = 1, \ldots, m. \tag{7}$$

For each product term, there are two new constraints and one new integer variable.

The second method, described in Glover [1975] for quadratic terms, can be used for higher-order terms by recursive application (Stecke [1981]). For m x m quadratic terms, Balas's approach (method 1) introduces m(m-1)/2 new integer variables (one for each cross-product term) and m(m-1) additional constraints. Glover's approach adds 4m constraints and m continuous variables, which are automatically 0-1 without requiring an integer restriction. The second method has the advantage that the transformed linear integer program has the same number of integer variables as the original nonlinear program.

The third method (Glover and Woolsey [1974]) allows the new variables $x_{\bar{S}}$ (of equations (6) and (7)) to be continuous by replacing the second inequality of Balas's method (equation (7)) with $|\bar{S}|$ additional constraints. Despite the additional constraints, method 3 can be better than the first method since the additional constraints are simpler and the new variables are continuous.

The fourth method (Glover and Woolsey [1974]) also allows $x_{\bar{S}}$ to be continuous by replacing several of the constraints (7) that contain terms with common variables with a single constraint. If there are no variables in common, there is no reduction.

The final method (Glover and Woolsey [1973]) consists of a series of three rules that reduce the number of required constraints of the first of each pair of constraints (equation (6)) generated by Balas's method. The new variables are continuous.

4. Application and Computational Results

Combinations of the five linearizing methods are applied to data from the Sundstrand DNC (Direct Numerical Control) line at the Caterpillar Tractor Company in Peoria, Illinois. The Caterpillar FMS consists of nine metal-cutting machines plus an inspection station. This set of machines includes four 5-axis Omnimills, three 4-axis Omnidrills, and two vertical turret lathes (VTLs). In addition, there are two automatically controlled transporters, which provide in-process material handling and also deliver parts to the inspection station. The 16-station load/unload area is located midway along the line's length. These stations also provide a queueing area for in-process inventory. A remotely located DEC PDP 11/20 computer and supporting equipment control the entire system on a real-time basis.

The parts machined on this line are two sizes of housings for automatic transmissions. Each type of housing is composed of two parts, a transmission case and a cover. The parts arrive at the facility in rough casting form and leave as an assembled matched pair. There are three part types: transmission cases, covers, and assemblies.

Caterpillar's loading objective was to balance the assigned workload per machine as much as possible; in addition, each operation was assigned to only one machine.

The machines have different capabilities. Some operations require a mill; others can be performed on either a mill or a drill (mills can do drilling operations, but not vice versa). The two VTLs could be pooled, although they were not in management's original set-up strategies. After certain operations, a proportion of the parts will visit the inspection station. For additional information concerning the management and control of the DNC line, see Stecke [1977].

4.1 Input

Some operations are collected in advance into operation sets. For example, a large case requires 49 operations (Stecke [1977]), which Caterpillar had aggregated into nine operation sets. These operation sets, along with those of covers and assemblies, are allocated among machines according to various loading objectives.

The input data includes, for each operation set, the machine type required; the total number of tool slots required; the tool number and number of slots for any tool of that operation set which is required by at least one other operation set; and the processing time.

Initial calculations include a table of the number of tool slots saved (w_{B_k}). This table, as well as the constraint formulations, are found in Stecke and Solberg [1981].

4.2 Constraints Linearized and Compared

The tool magazine capacity constraint is formulated, and then linearized according to the different methods. The best (combination of) method(s) that generated the fewest additional variables and/or additional constraints was to be run on a CDC 6600 in conjunction with each loading objective and the grouping objective.

The basic nonlinear formulation consists of 48 integer variables and 25 constraints (see Table 3, which also contains the number of additional integer (continuous) variables and constraints that are generated by each of six combinations of the five linearizing methods). The new variables that are in parentheses are automatically 0-1 when the original variables are constrained to be so. Application of each of the first two methods results in very different constraint linearizations. Methods 3 and 4 replace the second of each pair of constraints generated by method 1, while the fifth method replaces the first constraint of each pair of method 1.

TABLE 3

Basic Number of Variables and Constraints
Plus Those Generated by Linearizing

	Basic Nonlinear Formulation		Linearization Methods 1	2	3	4	5	4,5
Variables Integer (Continuous)	48	+	113	(76)	(113)	(113)	113	(113)
Constraints	25	+	218	228	373	157	152	91

There are two sets of linearized constraints that are candidates for selection to use for solving the MIPs: (1) method 2 and (2) methods 4 and 5. From Table 3, we see that method 2 generates fewer variables, and methods 4 and 5 generate fewer constraints. Since the difference in the number of constraints is large (127 fewer--over 58% reduction), and the new variables are all continuous, the constraint set chosen to run the MIPs is that generated by the fourth and fifth methods.

4.3 Comparison of Linearizations

Each linearization method is applicable to different types of nonlinear problems. If the product of nonlinear 0-1 variables is quadratic, then the second method (Glover [1975]) may be best, since it results in fewest additional continuous variables and no new integer. Since higher-order product terms are present for our problem, the second linearization procedure must be applied iteratively to produce an increasing number of generated constraints, which is why the combination of methods 4 and 5 was chosen. In similar situations, the combination of methods 4 and 5 would usually be better than method 2.

In general, if there is a set of integer problems to be solved in which the problems have different, higher-order, nonlinear product terms in either the constraints or the objective functions, method 2 would not be best, for the following four reasons. We claim that the definition of new continuous variables from the higher-order terms is not, in general, unique. Hence, the generated constraint set is not unique. It is not clear a priori which generated set is best. Also, adding or changing nonlinear terms may cause a necessary relinearization of much of the problem, for it to be as efficient (fewest additional variables) as possible. Finally, the second method can require additional constraints that use variables that have already been linearized. None of the other methods will require additional constraints for these variables. Examples that demonstrate these claims can be found in Stecke [1981].

Although these guidelines are true in general, not all nonlinear integer problems demonstrate these properties. Problems that have a small constraint size, and problems in which the constraints contain few terms in common, cannot be reduced significantly. In these cases, method 2 would be best because the new variables would be continuous. An attempt to apply the other methods would result in: (1) few, or no, reductions in the constraint size, (2) a greater number of additional variables than method 2, and (3) new variables that would be integer rather than continuous.

Finally, these observations stem from a small problem set. Further testing should be done to specify, more precisely, the realm of applicability of each linearization.

4.4 Objective Functions Linearized

The first objective function formulated is the grouping objective, which maximizes pooling. To accomplish this, the number of machines required is minimized. The remaining machines can then be pooled as indicated in the Appendix.

The two machine types, n, are mills and drills. Let $m_{oM}(m_{oD})$ be the upper bound on the number of mills (drills) required when overlap is not considered. The m_{on} are obtained by rounding to the next highest integer the ratio of the number of slots of all operations that require machine type n (M or D) divided by t_n. Then

$$m_{oM} = \lceil 169/60 \rceil = \lceil 2.8 \rceil = 3 \text{ mills}$$

$$m_{oD} = \lceil 68/60 \rceil = \lceil 1.113 \rceil = 2 \text{ drills},$$

where $\lceil x \rceil$ denotes the least integer greater than or equal to x.

The grouping objective maximizes a linear combination of the slack variables to find the minimum number of required machines of each type. There are no additional constraints or variables that have not already been linearized for the capacity constraints in §4.2.

The second objective formulated minimizes movements. The number of additional constraints required when using the second linearization method is 48, and there are 16 new continuous variables. The first method introduces 16 integer variables and 32 constraints. However, several of the variables have previously been defined and linearized for the capacity constraints. Hence, the first method introduces only 9 new integer variables and 16 new constraints. This information is summarized in Table 4. The constraint set cannot be reduced further by methods 4 or 5 because no pair of constraints contains any common variables.

The next loading objective that is linearized balances the assigned machine processing times. Some inequalities were used to reduce the number of new variables and constraints. The basic formulation summarized in Table 4,

TABLE 4

Objective Function Linearizations

		Nonlinear Formulations		Linearization Methods 1	2	
Minimize Movements	Variables	-	+	9	(16)	Constraint set cannot be reduced further
	Constraints	-	+	16	48	
Balancing	Variables	(7)	+	(12)	-	
	Constraints	7	+	29	-	

introduces 7 continuous variables (the r_j) and 7 constraints. Linearization by method 1 adds 12 continuous variables and 29 constraints. Method 2 introduces no additional variables or constraints.

Finally, note that a composite objective can be defined that simultaneously minimizes movements as well as the number of required machines. This is achieved using a linear combination of the two objectives.

Formulations of the third (and fourth) loading objectives, (un)balancing, are similar to that of the first objective. In addition, the resultant formulations are smaller than the first. Since machines are partitioned into groups and assignments are identical for each machine in a group, a capacity constraint is required only for each group rather than for each machine. These remaining smaller formulations are not linearized and solved here.

4.5 Effect of the Linearizations on Problem Sizes

A summary of the sizes of the linearized MIP formulations of the grouping problem and the two representations of the loading problem is given in Table 5. From Tables 3 and 5 we can conclude that the application of methods 4 and 5 significantly reduced the constraint size of the set generated by method

TABLE 5

Problem Sizes

	OBJECTIVES		
	Maximize Pooling	Minimize Movements	Balancing
Variables	48 + (113) = 161	57 + (113) = 170	48 + (132) = 180
Constraints	116	132	152

1, by 127 constraints, resulting in 116, 132, and 152 constraints, respectively, for the three problems. In addition, nearly all of the new variables for each of the three problems (113 out of 113, 113 out of 122, and 132 out of 132) were continuous.

The computer code used to solve the three linearized mixed integer programming problems, MIPZ1, is described in McCarl, Barton, and Schrage [1973].

Solution times ranged from about 1.5 to 2.5 minutes on a CDC 6600. The code is an adaptation of the code developed by Bravo et al. [1970] and requires integer variables to be either zero or one. The algorithm is a modification of Balas's Additive Algorithm [1965] along the lines suggested by Glover [1968] and Salkin [1970]. Details are described in McCarl et al. [1973].

4.6 Solution Quality

The nonlinear tool magazine capacity constraints resulted in larger linear MIPs, but also in better solutions. The solution to the grouping problem in the example was that all three drills could be pooled. However, if tool overlap and duplication were ignored, the solution is that two drills are needed to hold all required tools (see §4.4). Consideration of tool duplication also allowed more pooling of mills than otherwise.

The optimal solutions of the three MIPs are identical to those in Stecke and Solberg [1982b], which were obtained by heuristic means according to the same balancing/minimizing movements/pooling objectives.

5. Summary and Conclusions

In this paper, we addressed the general problem that an FMS manager has in setting up his job shop-like system for efficient production. To this end, we presented a conceptual framework within which the set-up problem may be viewed as a series of five production planning problems. We then mathematically defined and solved two of these problems. The detailed nonlinear MIP formulations of the grouping and loading problems provide optimal solutions that are useful in actual applications. Linearizing methods are suggested as one approach to solving these problems. We claim that each linearization is appropriate for a different type of problem. However, additional problems should be examined to further clarify the extent of applicability of each linearization.

The linearized MIP formulations were applied to data from an existing FMS. Although the results are not definitive, they are encouraging, since the grouping and several loading problems were solved optimally using a standard MIP code. Therefore, the linearized MIPs can be solved at least for common problem sizes. For larger systems, additional research should be done. Particular MIP codes that exploit the special structures found in these problems can be developed. Heuristic procedures, piecewise linear approximations, or fast approximation algorithms could perhaps be used if optimal loadings are not required. Finally, a linear relaxation can be used, with a heuristic post-adjustment of the solution to eliminate operation splitting.[1]

[1] The author would like to thank Bruce W. Schmeiser and James J. Solberg of Purdue University, as well as Fred Glover of The University of Colorado and F. Brian Talbot of The University of Michigan, for their helpful comments during the preparation of this manuscript. The research was supported in part by the National Science Foundation Grant No. APR74 15256. The author also thanks the referees for their suggestions, which improved the presentation of the results.

Appendix: Theoretical Grouping and Loading Results

The theoretical results (Stecke and Solberg [1982a]) that are used here in §2 are now summarized. These results were obtained through the use of a closed queueing network model, which represented an FMS.

Assume that there is a system of m machines, M machine groups, N parts, s_ℓ machines in group ℓ, and that X_ℓ is the workload assigned to group ℓ. The groups are ordered according to increasing size, that is,

$$s_1 \leq s_2 \leq \cdots \leq s_M.$$

Define $\max_X \Pr(M,N,(s_1,\ldots, s_M);X)$ to be the maximum expected production from the system, where $X = [X_1, \ldots, X_M]$.

Then for any integer $K > 0$, we have that

1) $\operatorname{maximum}_X \Pr(M,N,(s_1,\ldots,s_i+K,\ldots,s_M-K);X)$ is strictly less than $\operatorname{maximum}_X \Pr(M,N,(s_1,\ldots,s_i,\ldots,s_M);X)$;

2) $\Pr(M,N,(s_1,\ldots,s_i+K,\ldots,s_M-K);\vec{1})$ is strictly greater than $\Pr(M,N,(s_1,\ldots,s_M);\vec{1})$;

3) $\Pr(M,N,(s_1,\ldots,s_i,\ldots,s_j,\ldots,s_M);X)$ is strictly greater than $\Pr(M,N,(s_1,\ldots,s_i+K,\ldots,s_j',\ldots,s_M);X)$, for $s_i+K=s_{i+1}+K=\ldots=s_j'$.

The order of groups according to size is preserved.

The first and third statements say that the maximum expected production increases when system configurations are unbalanced. The second statement indicates that equalizing group sizes decreases maximum expected production.

To clarify, the ordering of all partitions of m machines into three groups to maximize expected production is as follows, with the best grouping first.

$$
\begin{gathered}
(1,\ 1,\ m-2) \\
(1,\ 2,\ m-3) \\
\vdots \\
\begin{pmatrix} 1,\ (m-1)/2-1,\ (m+1)/2, \text{ if } m \text{ is odd;} \\ 1,\ m/2-1,\ m/2, \quad \text{ if } m \text{ is even.} \end{pmatrix} \\
\vdots \\
\begin{pmatrix} m/3,\ m/3,\ m/3, \text{ if } 3 \text{ divides } m; \\ (m-1)/3,\ m/3,\ m/3, \text{ if } 3 \text{ divides } m+1; \\ (m-1)/3,\ (m-1)/3,\ m/3, \text{ if } 3 \text{ divides } m+2. \end{pmatrix}
\end{gathered}
$$

The more unbalanced partition is capable of the larger maximum expected production.

The theoretical loading problem is to allocate a total amount of work among a system of grouped machines to maximize expected production. The solution is a set of optimal allocation ratios, $X_1^*, \ldots, X_M^*$, or theoretical ratios at which the groups should be assigned work.

In particular, the solution to the theoretical loading problem is as follows: The expected production is maximized by assigning

i) a balanced workload to each machine, if all group sizes are identical;

ii) a specific, unbalanced workload to each machine, if group sizes are unequal.

REFERENCES

1. BALAS, EGON, "Extension de l'algorithme additif à la programmation en nombres entiers et à la programmation non linéaire," C.R. Acad. Sc. Paris (May 1964).

2. BALAS, EGON, "An Additive Algorithm for Solving Linear Programs with Zero-One Variables," Operations Research, Vol. 13 (1965), pp. 517-545.

3. BARASH, MOSHE M., "Speculation on the Future of Numerical Controls," paper 78-WA/DSC-9 presented at the American Society of Manufacturing Engineers Conference, San Francisco CA (December 1978).

4. BARASH, MOSHE M., "Computerized Manufacturing Systems for Discrete Products," Ch. VII-9 in The Handbook of Industrial Engineering, Gavriel Salvendy, (ed.), John Wiley & Sons, New York, 1982, forthcoming.

5. BERDINE, ROBERT A., "Caterpillar's DNC System 2-1/2 Years Later," Proceedings of the 15th Numerical Control Society Annual Meeting and Technical Conference, Chicago IL (April 9-12, 1978), pp. 110-115.

6. BRAVO, A., GOMEZ, J.G., LUSTOSA, L., SCHRAGE, L. and PIZZOLATO, N.D., A Mixed Integer Programming Code, Report 7043, University of Chicago, Chicago IL, September 1970.

7. COOK, NATHAN H., "Computer-Managed Parts Manufacture," Scientific American, Vol. 232, No. 2 (February 1975), pp. 22-28.

8. COOPER, MARY W., "A Survey of Methods for Pure Nonlinear Integer Programming," Management Science, Vol. 27 (1981), pp. 353-361.

9. DANTZIG, GEORGE B., Linear Programming and Extensions, Princeton University Press, Princeton NJ, 1962.

10. DIRKSEN, GORDEN F., "Software Components for Multi-Station, Digitally Controlled Manufacturing Systems," presented at a workshop held at the University of Wisconsin, Milwaukee WI, published as a report (January 1977).

11. GINSBURGH, V. and VAN PEETERSEN, A., "Un algorithme de programmation quadratique en variables binares," Réview Francoise d'Informatique et de Recherche Operationnelle, Vol. 3 (1969), pp. 57-64.

12. GLOVER, FRED, "Surrogate Constraints," Operations Research, Vol. 16 (1968), pp. 741-749.

13. GLOVER, FRED, "Improved Linear Integer Programming Formulations of Non-Linear Integer Problems," Management Science, Vol. 22 (1975), pp. 455-460.

14. GLOVER, FRED, "Heuristics for Integer Programming Using Surrogate Constraints," Decision Sciences, Vol. 8 (1977), pp. 156-166.

15. GLOVER, FRED and WOOLSEY, EUGENE, "Further Reduction of Zero-One Polynomial Programming Problems to Zero-One Linear Programming Problems," Operations Research, Vol. 21 (1973), pp. 156-161.

16. GLOVER, FRED and WOOLSEY, R.E., "Converting the 0-1 Polynomial Programming Problem to a 0-1 Linear Program," Operations Research, Vol. 22 (1974), pp. 180-182.

17. HAMMER, PETER L., "A B-B-B Method for Linear and Nonlinear Bivalent Programming," Operations Research, Statistics and Economics Mimeograph Series No. 48, Technion, May 1969.

18. HANSEN, P., "Nonlinear 0-1 Programming by Implicit Enumeration," paper presented at the VII Mathematical Programming Symposium, The Hague, September 1971.

19. HANSEN, P., "Methods of Nonlinear 0-1 Programming," Annals of Discrete Mathematics, Vol. 5 (1979), pp. 53-70.

20. HOCHBAUM, DORIT, "Fast Aproximation Algorithms for Some Integer Programming Problems," Working Paper #21-80-81, Carnegie-Mellon University, Graduate School of Industrial Administration, Pittsburgh PA 15212, October 1980.

21. HU, T.C., Integer Programming and Network Flows, Addison-Wesley, Reading MA, 1969.

22. LAWLER, EUGENE L. and BELL, M.D., "A Method for Solving Discrete Optimization Problems," Operations Research, Vol. 15 (1967), pp. 1098-1112.

23. MARSTEN, ROY E. and MORIN, THOMAS L., "A Hybrid Approach to Discrete Mathematical Programming," Mathematical Programming, Vol. 14 (1978), pp. 21-40.

24. McCARL, BRUCE, BARTON, DAVID and SCHRAGE, LINUS, "MIPZ1--Documentation on a Zero-One Mixed Integer Programming Code," Dept. of Agricultural Economics, Purdue University, W. Lafayette IN, September 1973.

25. MORIN, THOMAS L., "Computational Advances in Dynamic Programming," Dynamic Programming and Its Applications, Martin L. Puterman (ed.), Academic Press, New York, 1978.

26. SALKIN, HARVEY M., "On the Merit of Generalized Origin and Restarts in Implicit Enumeration," Operations Research, Vol. 18 (1970), pp. 549-555.

27. STECKE, KATHRYN E., "Experimental Investigation of a Computerized Manufacturing System," Master's Thesis, Purdue University, W. Lafayette IN 47907, December 1977.

28. STECKE, KATHRYN E., "Production Planning Problems for Flexible Manufacturing Systems," Ph.D. Dissertation, Purdue University, W. Lafayette IN 47907, August 1981.

29. STECKE, KATHRYN E. and SOLBERG, JAMES J., "The CMS Loading Problem," Report No. 20, NSF GRANT No. APR74 15256, School of Industrial Engineering, Purdue University, W. Lafayette IN 47907, February 1981.

30. STECKE, KATHRYN E. and SOLBERG, JAMES J., "The Optimality of Unbalanced Workloads and Machine Group Sizes for Flexible Manufacturing Systems," Working Paper No. 290, Graduate School of Business Administration, The University of Michigan, Ann Arbor MI 48109, January 1982a.

31. STECKE, KATHRYN E. and SOLBERG, JAMES J., "Loading and Control Policies for a Flexible Manufacturing System," International Journal of Production Research, 1982b, in press.

32. TAHA, HAMDY A., "A Balasian-Based Algorithm for 0-1 Polynomial Programming," Research Report No. 70-2, University of Arkansas, Fayetteville AR, May 1970.

33. WATTERS, LAWRENCE J., "Reduction of Integer Polynomial Programming Problems to Zero-One Linear Programming Problems," Operations Research, Vol. 15 (1967), pp. 1171-1174.