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

None of this material is to be quoted or reproduced without the express permission of the Division of Research.
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
To maximize expected production, the objective of a set-up is system-dependent, but commonly it is reserved for each part type and allocation of operations (and tools) to simultaneously satisfy number of pallets and fixtures of each fixture type to be produced, next, relative numbers of parts of each type to be machined.

The decision variables of the set-up problem of an RMS are:

- Part types
- Changes are few

A mass production system where set-up is part of the design process and set-up and production are prioritized to meet new or altered production requirements for a part type are met. This contrasts with frequent setup to meet new or altered production requirements, for example, for prior to production. Set-up decisions for batch manufacturing are made up to utilize an RMS's capabilities, a careful system set-up is required.

The size of the system, the number of decision variables, and the constraints associated with setting the system's capabilities and planning options increase both types simultaneously, and each part may have alternative routes through the system. The system can machine several parts performing many different operations. The system is able to operate at nearly full capacity and is capable of maintaining production for an RMS is more difficult than for production.

(Dirksean [1977])

Planning is needed by the initial RMS users (Berdal [1978]) and manufacturers still in their infancy, problems have been encountered. The need for precise because the concepts and technology of automated batch manufacturing are for planning prior to production. Since last requirement is placed in the machine, this last requirement determines the need for machine setup and tool interchanges. The computer cannot route a particular operation to be machined with virtually no set-up time between operations.
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 grouping and loading. For additional information concerning all of these problems, see Stecke [1981].
perform the same operations. Hence, each machine in a particular group will be able to
satisfy the same owners. Machines that are inherently similar can be grouped and are
designed to be pooled. But, each set of machine types, with similar part operation into several man-
ufactured, are designed to be machine type a's. Machine types of the same machine type are physically identical.

The differences between machine types and machine groups should be

subsection 4 is needed.

Table 4 shows which type is only one machine in each group. The machine
matched another to either machine or machine groups. In the formation of the parameters given in Table 1
explanation. In the formation of several of the parameters given in Table 4.

Table 4. Several subsections concerning and parameters required further
The subsections, input parameters, and decision variables are given in

Parameters and Variables

Objectives are developed.

Grouping problem is then defined and formulated. Finally, several loading
constraints are necessary for the grouping and loading problem. After defining the grouping, this section begins by developing the

2. Mathematical Programming Formulations

nonlinear terms. § 4 presents a summary and conclusion.

computational experience, which shows the advantages of considering the
under which each interaction is best. § 4 also includes a discussion of
and/or variables. In addition, conclusions are drawn concerning conditions
using the interaction that results in the greatest additive constraints
and/or variables. § 5 presents several interaction methods. In § 4, these are
problems (MIPs). In addition, different methods of solving nonlinear MIPs

The plan of the paper is as follows. The machine grouping (2) and load-

-4-
TABLE 1

Notation

Subscripts:

operation \( i = 1, \ldots, b \)
machine \( j = 1, \ldots, m \)
machine group \( \ell = 1, \ldots, M \)
machine type \( n = 1, \ldots, \bar{m} \)
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 \( \ell \)

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

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

\( t_{\ell} = \) capacity of the tool magazine for each machine in group \( \ell \)

\( w_{ik} = \) 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

\( \bar{B} = \) index subset of \( B_k \) such that \( |\bar{B}| \), the cardinality of \( \bar{B} \), \( = p \), \( p = 2, \ldots, b \)

\( w_{B_k} = \) 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):

\( M_{\ell} = \{ j \mid \text{machine } j \text{ is in machine group } \ell \} \)

\( x_{i\ell} = \begin{cases} 
1, & \text{if operation } i \text{ is assigned to each machine in group } \ell; \\
0, & \text{otherwise.}
\end{cases} \)
plcation and considering overlap and weight balancing. These savings are
some tools. Space in the tool magazine can be saved by eliminating tool
weight balances. In addition, several operations may require some of the
storages. It is that since lager tools are heavier, tool magazines must be
by side require only five stores rather than six. Another complicating con-
 vaginal than Figure 1. In the example shown in Figure 1, two three-slot tools placed side
of stores used depends on the physical placement of the tools in the tool
assign any tool more than once to the same machine. Also, the actual number
since only one tool can be used at a time, however, it is unnecessary to

\[ I = T_{x} \sum_{p} I_{x} P \frac{I}{q} \]

is the simplest form.

First, the total number of stores assigned in the machine (group)’s tool magazine, in
the machine type corresponding to machine (group) \( f \) or \( g \).

It is understood that if operation \( j \) cannot be performed by the

\[ I = T_{x} \sum_{p} I_{x} P \frac{I}{q} \]

on the number of duplicate assignments allowed.

The constraints of the group and planning problems are as follows:

Constraints Formulations
measured by \( w_{B_k} \). The tool magazine capacity constraint then becomes:

\[
\sum_{i=1}^{b} d_i x_{i,k} - \sum_{i_1=1}^{b-1} \sum_{i_2=i_1+1}^{b} w_{i_1,i_2} x_{i_1,k} x_{i_2,k}^+ + \\
+ \sum_{i_1=1}^{b-2} \sum_{i_2=i_1+1}^{b-1} \sum_{i_3=i_2+1}^{b} w_{i_1,i_2,i_3} x_{i_1,k} x_{i_2,k} x_{i_3,k}^+ + \ldots \\
+ (-1)^{p+1} \sum_{i_1=1}^{b-p+1} \sum_{i_2=i_1+1}^{b-p} \ldots \sum_{i_p=i_{p-1}+1}^{b-p+2} w_{i_1,\ldots,i_p} x_{i_1,k} \ldots x_{i_p,k} \leq t_k,
\]

or, in more compact form,

\[
\sum_{i=1}^{b} d_i x_{i,k} + \sum_{p=2}^{b} (-1)^{p+1} \sum_{i_1=1}^{b} w_{B_p} \prod_{k=1}^{p} x_{i_k,k} \leq t_k, \quad k = 1, \ldots, M. \quad (2)
\]

Finally, there is the integrality constraint:

\[
x_{i,k} = 0 \text{ or } 1, \text{ for all } i, k. \quad (3)
\]

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.
The objective function formulation allows for the values of the slack

\[
\begin{align*}
\min \quad & \sum_{w} \frac{1}{w} \cdot I_{w} = 0 \\
I_{w}^{=} & = 0 \\
\end{align*}
\]

where \( w \) is the number of machines of machine type \( w \). Then

the tool required of machine type \( u \) is determined by the capacity of each machine tool.

Let \( w \) be the maximum number of machines of machine type \( w \). Then

the stack in the tool magazine capacity constraint of machine \( j \) is merely a Lagrange number, so that \( I \) is not a Lagrange multiplier.

\[
\begin{align*}
I & = I_{d} + I_{a} + I_{w} + \frac{1}{\eta} \left( \sum_{v} \frac{d_{v}}{w_{v}} \right) + \sum_{l=1}^{d}(I-l) \left( \sum_{p} \frac{1}{q_{p}} \right) + \sum_{q=1}^{d}(I-q) \left( \sum_{p} \frac{1}{q_{p}} \right) - f \frac{1}{\eta} \\
\end{align*}
\]

subject to

\[
\begin{align*}
\sum_{w} I_{w}^{=} & = I \\
\end{align*}
\]

maximize

\[
\begin{align*}
\min \quad & \sum_{w} \frac{1}{w} \cdot I_{w} = 0 \\
\end{align*}
\]

Step 1, which determines \( N \), is now described. The problem is to find

more detailed model than the best feasible partition.

Since all possible partitions of machines are ordered, the solution to the

(provided in the Appendix)

\[
\begin{align*}
2. \quad & \text{Use the optimal partition for } N \text{ groups from Stecke and Solberg [1982]} \\
\end{align*}
\]

required to perform all operations of the part types in \( P \). I.e., \( N \) equal to the minimum number of machines (of machine groups)

\*

The approach we take to maximize pooling is as follows:

be possible because of some technological constraints such as tool magazine

Maximum pooling of all machines of the same type into one group may not

\*

-8-
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 re-
quired 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:

\[
\begin{array}{cccc}
  n: & 1 & 2 & 3 & 4 \\
  machine \ j: & (1 \ 2 \ 3 \ 4) & (5 \ 6 \ 7) & (8 \ 9 \ 10 \ 11 \ 12) & (13 \ 14 \ 15)
\end{array}
\]

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:

\[
\begin{array}{cccc}
  machine \ j: & (1 \ 2) & (3) & (4) & (5) \ (6) \ (7) \ (8 \ 9 \ 10) \ (11) \ (12) \ (13 \ 14 \ 15)
\end{array}
\]

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 pro-
duction 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
Each objective function is a measure of system imbalance. Suppose optimally balance the workloads assigned to machines as much as possible. Suppose because of the discrete values of processing times, the following for the if are equal, the system is perfectly balanced. This is usually not possible. Then it is also a measure of the relative utilization of machine i. If all of the processing times, let i be the relative workload assigned to machine i.

Consider the first objective function, balancing the assigned machine

1. Maximize the sum of operation priorities.
2. Maximize the tool availability as closely as possible.
3. Pile the tool availability of unequal sizes.
4. Balance the workload per machine for a system of groups of
   pooled machines of unequal sizes.
5. Balance the workload per machine for a system of groups of
   pooled machines of equal sizes.
6. Balance the workload per machine for a system of groups of
   operations on each machine.
7. Maximize the number of movements from machine to machine.
8. Maximize the number of movements between machines.

Loading Objectives

| TABLE 2 |

Applicable

Table 2 concentrates while in others, several objectives may be equally best under certain circumstances. In some situations, some of the objectives are equally best under certain circumstances which to apply is problem-dependent. Each may be of system performance. Several alternative loading objectives are listed in Table 2. More precise, since the inherent extensibility can often be utilized for better performance than attempting to balance assigned workloads.

Application of any of several objectives can result in better system performance. Previous study (Stecch [1977]), alternative loading objectives were defined.
The problem is to find \( \{x_{ij}\} \text{ i=1, ..., b, j=1, ..., m} \) to minimize \( h_i(x) \), where \( h_i(x) \) is one of the following four:

1. \[
\text{maximum } \sum_{j=1, \ldots, m-l} \sum_{h=l}^{m-1} |r_j - r_h| \]

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

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

4. \( \beta - \alpha \),

subject to \( 0 \leq \alpha \leq r_j \leq \beta, \quad j = 1, \ldots, m \).

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

\[
x_{ij} - x_{i+1,j} = \begin{cases} 
0, & \text{if operations } i \text{ and } i+1 \text{ are on the same machine } j; \\
1, & \text{if operations } i \text{ and } i+1 \text{ are assigned to different machines.}
\end{cases}
\]
Notice that if \( r \) is the workload per machine in machine group \( g \),
subject to (1), (2), (3), (4), and (5), then (6), (7), (8), (9), (10), and (11).

Subject to \( g \), \( r \), and \( h \),

\[
\begin{align*}
I = \frac{\gamma S}{\gamma L} & \quad \gamma = \frac{I + \gamma = 1}{I - \gamma} \\
\frac{\gamma S}{\gamma L} & \quad \gamma = \frac{I = \gamma}{I - \gamma} \\
0 < \lambda & \quad \lambda = \frac{\gamma S}{\gamma L} \\
\max_{I} & \quad \gamma = \frac{I = \gamma}{I - \gamma} \\
\end{align*}
\]

is to minimize \( h(x) \), where \( h(x) \) is one of the following four

is the objective function. For the third objective, the problem

either depends on the configuration of the manufacturing systems, or how the

machine for a system of groups of pooled machines. The applicability of

machine to the (third and fourth) objectives are to (un)balance the workload per

takes. The third (and fourth) objectives are to obtain balance between the remaining

alternative part routes provide motivation for the remaining loading object-

The advantages from utilizing flexibility by allowing pooling and hence

Second objective, then, is to

N. Inclusion is not incorrect, merely unnecessary and inefficient. The
case, \( \gamma = 0 \). In part-

calculation of \( N \). For example, for some machine \( j \) operate \( \gamma \) may require a

some of the differences between the number of excess movements, then

\[ (I + I)^{\gamma} - (I^\gamma) \]

This may seem odd, but this term may be excluded from

The fourth problem is to minimize $g_4(x)$, where $g_4(x)$ is one of the following four:

1. $\max_{\ell=1, \ldots, M} \left| r_\ell - X_\ell^* \right|$

2. $\max_{\ell=1, \ldots, M} \left| r_\ell - X_\ell^* \right| \gamma, \gamma > 0$

3. $\sum_{\ell=1}^{M} (r_\ell - X_\ell^*)^2$

4. $\beta - \alpha,$
   subject to $0 \leq \alpha \leq r_\ell - X_\ell^* \leq \beta, \quad \ell=1, \ldots, M,$
   subject to (1), (2), (3), (5), and $q_4 = 1,$ and where $X_\ell^*$ is the theoretical optimal workload that should be assigned to machine group $\ell$ 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

$$\text{minimize} \quad \sum_{j=1}^{m} s_j^2$$

subject to (1), (3), $q_4 \geq 1,$ and

$$s_j^2 = t_j - (\sum_{i=1}^{b} d_i x_{ij} + \sum_{p=2}^{b} (-1)^{p+1} \sum_{W_B \subseteq B_k} \prod_{i_k \in B_k} x_{i_k j}).$$

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

### 2. Linearization of the Product Terms

Nonlinear terms, combinations of these methods are applied to data in §4.4.
In the following section, we present several methods to linearize the

Algorithms (1980)½

Linearized integer problems can be solved by means of fast approximation
difficult, while linearization results in much larger problems. Finally, the
approach is problem dependent (Thomas 1970)½. The direct approach is more
applicable if a direct nonlinear approach versus a transformed linear
linearized 0-1 problems could either be solved directly or relaxed. The
resultant solution is to linearize the nonlinear terms (Balas 1964)½, Glover and Woolsey 1974, 1975)½. The resultant
solutions. An exact approach to linearize the nonlinear terms (Balas
capacity constraints. Ignoring these factors results in feasible, but worse,

Reference some toolsing constraints that result in the nonlinear terms in the
Stecke and Solberg 1981)½ can be used. Another approximate approach is to

transformations (masker and Belli 1977)½. Alternatively, proceed via linear approximation

They can be solved directly (Cooper 1980)½, Charnley and van Pelter
nonlinear. A variety of approaches for solving nonlinear MIP is available,

Almost all of the objective functions and some of the constraints are
subject to (1) and (2)

\[
\begin{align*}
& \text{maximize } \sum_{i} \sum_{j} q_{i,j} \times x_{i,j} \\
& \text{subject to } \sum_{j} x_{i,j} = f_i \\
& \end{align*}
\]

As opposed to operation \( I \), then the problem is to

Could be subjected to predefined operation assumptions? If \( I \) is the weight
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 \( \prod_{i_k \epsilon S} x_{i_kj} \) is replaced by a new variable \( x_S^{\pi} \)

\[
\sum_{i_k \epsilon S} x_{i_kj} - \pi x_S^{\pi} \leq p - 1, \quad p = |S| \tag{6}
\]

\[
\sum_{i_k \epsilon S} x_{i_kj} + p x_S^{\pi} \leq 0, \quad j = 1, \ldots, m. \tag{7}
\]

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 \( m \times m \) 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 \( x_S^{\pi} \) (of equations (6) and (7)) to be continuous by replacing the second inequality of Balas's method (equation (7)) with \( |S| \) 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 \( x_S^{\pi} \) to be continuous by replacing several of the constraints (7) that contain terms
The machines have different capabilities. Some operations require a
one-machine.

to be performed on either a milling or a drilling machine. Others can be performed

The case, covers, and assemblies are composed of two parts, a housing and
a transmission. Each type of housing is composed of two parts, a transmission
and a cover. The parts arrive at the facility to rough casting form and
leave as an assembled matched part. There are three part types: transmission,
inventory, and supporting equipment. A remote facility located off line/remote
controls the entire system on a real-time basis. A control computer and supporting

The 16-station load/unload area is located along the
inspection station, which provides in-process material handling and also delivers the parts to the

In addition, there are two automated transfer controlled transporters,
4-axis compound, and two vertical turret lathes.

The carretpillar's loading objective was to balance the assigned workload per
Caterpillar's loading objective was to balance the assigned workload per

case, covers, and assemblies.
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
than method 2.

smaller situations, the combination of methods 4 and 5 would usually be better

comparisons, which is why the combination of methods 4 and 5 was chosen. In
must be applied iteratively to produce an increasing number of generated
product terms are present for our problem, the second linearization procedure
addition, continuous variables and no new integer. Since higher-order
second method (Clover [1979]) may be best, since it results in least
problems. If the product of nonlinear 0-1 variables is quadratic, then the
the linearization method is applicable to different types of nonlinear

4.3 Comparison of Linearizations

Generated by the Fourth and Fifth Methods.

are all continuous, the constraint set chosen to run the MIP is that
constraint is large (127 fewer—over 56% reduction), and the new variables
constraint is fewer again. Since the difference in the number of
and 5 generate fewer constraints. Since the difference in the number of
5. From Table 3, we see that method 2 generates fewer variables, and methods
selection to use for solving the MIP: (1) method 2 and (2) methods 4 and
there are two sets of linearized constraints that are candidates for

| 91 | 127 | 157 | 373 | 228 | 218 | + | 25 |
|----|-----|-----|-----|-----|-----|+|----|
| (I) | (I) | (I) | (I) | (I) | (I) | (I) | + | (4) |
| 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Table 3. These Generated by Linearizing
Basic Number of Variables and Constraints

<table>
<thead>
<tr>
<th>Linearization Methods</th>
<th>Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercritions</td>
<td>Cont.cntrns</td>
</tr>
<tr>
<td>Integer Variables</td>
<td>Formulation</td>
</tr>
<tr>
<td>Nonlinear</td>
<td>Nonlinear</td>
</tr>
</tbody>
</table>

TABLE 3

replaces the linear constraints of each pair of methods. I.

very different constraints. Linearizations. Methods 3 and 4 replace the second

-18-
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.
TABLE 4

Objective Function Inheritance

<table>
<thead>
<tr>
<th>Constraint</th>
<th>Inheritance Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>11</td>
<td>11, 13</td>
</tr>
<tr>
<td>09/89</td>
<td>09/89, 09/72, 09/69</td>
</tr>
</tbody>
</table>

The next loading objective that is Inherited balances the assigned ma-

...
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 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].
Although the results are not definitive, they are encouraging. Since the linear MIP formulations were applied to data from an existing

application, the applicability of each formulation

additional problems should be examined to further clarify the extent of

literature is appropriate for a different type of problem. However,

are suggested as one approach to solving these problems. We claim that each

optimal solutions that are useful in actual applications. The literature methods

mathematically defined and solved two of these problems. The deted

were used as a series of the production planning problems. We then

were presented a conceptual framework within which the set-up problem may be

in setting up his job shop-like system for efficient production. To this end,

in this paper, we addressed the General problem that an MIP manager has

5. Summary and Conclusions

same balancing/machining movements/tooling objectives

and some [1979] [1981b] which were obtained by heuristic means according to the

The optimal solutions of the three MIPs are heuristic to those in the

than that allowed more positioning of tips than otherwise.

needed to hold all required tools (see §4.4). Consideration of tool duplication

overlap and duplication were ignored, the solution is that two drill tips are

problem in the example was that all three drill tips could be pooled. However, if

linear MIPs, but also in better solutions. The solution to the group

the multiple tool machining capacity constraints resulted in larger

4.6 Solution Quality

* 1979 * [1968] and Sakin [1970] [1979a] * because are detected in Macket et al. * 1979 [1965] and Sakin [1970] [1979a] * because the times suggested by Grover integer variables to be either zero or one. The algorithm is a modification

is an adaptation of the code developed by Brevo et al. * 1970 [1979a] and requires

solution times ranged from about 1.5 to 2.5 minutes on a CDC 6600. The code
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 15296. 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_\(x\) machines in group \(x\), and that \(X_\lambda\) is the workload assigned to group \(\lambda\).

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) \(\max_X \Pr(M,N,(s_1,\ldots,s_i+K,\ldots,s_M-K);X)\) is strictly less than \(\max_X \Pr(M,N,(s_1,\ldots,s_i,\ldots,s_M);X)\);

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

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 \(s_i+K=s_i+K=\cdots=s_j\).
The expected production is maximized by assigning...

In particular, the solution to the theoretical loading problem is as follows:

- First, the order of groups according to size is preserved.
- Second, when system contributions are unbalanced, the second statement.
- Third and last statements say that the maximum expected production increases when system contributions are unbalanced. The second statement.
- Induces that equalizing group sizes decreases maximum expected production.
REFERENCES


24. McJUIN, Bruce, BARTON, David, and SCHREIBER, Linus, MIP/1: A Documentation of an Interactive Decision Tool, Department of Agricultural Engineering, Purdue University, West Lafayette, IN, September 1973.


29. STECKE, Kathryn E. and SOLBERG, James J., "The CMS Loading Problem," Report No. 20, NSF Grant No. #74-15256, School of Industrial Engineering, Purdue University, West Lafayette, IN, July 1981.

