

A hierarchical approach to solving machine grouping and loading problems of flexible manufacturing systems

Kathryn E. STECKE

Graduate School of Business Administration, The University of Michigan, Ann Arbor Michigan, U.S.A.

Abstract: A flexible manufacturing system (FMS) is an integrated system of computer numerically controlled (CNC) machine tools, each having an automatic tool interchange capability, and all connected by an automated material handling system. One or more computers control most real-time functions.

Flexible manufacturing is realized to be an efficient alternative to conventional manufacturing that allows simultaneous machining of small to medium batches of a variety of part types. Parts can flow through the system in unit batch sizes. These systems typically machine five to forty different part types.

In managing these systems, technological requirements indicate that several decisions must be made prior to system start-up. With these requirements in mind, previous research has defined a set of production planning problems, providing a conceptual framework to aid an FMS manager in setting up his/her system to enable efficient production. Several approaches have been taken to solve several of these problems and we describe those here. The main focus in this paper is on only two of these planning problems, the machine grouping and loading problems. In brief, the *FMS machine grouping problem* is to partition the m_i machine tools of type i into g_i groups to maximize expected production, subject to FMS technological and capacity constraints. Machines in a group are identically tooled and hence can perform the same operations during production. The *FMS loading problem* is to allocate operations and associated tooling of a selected set of part types among the machine groups, according to some appropriate (system dependent) loading objective, also subject to technological and capacity constraints.

This paper ties some previous results together by suggesting a hierarchical approach to solve actual grouping and loading problems. Both problems are first defined at an aggregated level of detail and in the context of a queueing network model. At this level, much information is suppressed. However, the robustness of the model allows the application of the obtained theoretical results to a lower level in the hierarchy that considers all details of these problems. In addition, results obtained using the aggregate model can be used as input to the detailed models. Here, the grouping and loading problems are formulated in all detail as nonlinear integer programs, using all available and required information. The use of these models to solve realistic machine grouping and loading problems is then described. Finally, future research needs are suggested.

Keywords: Planning problems, flexible manufacturing systems, queueing networks, resource allocation

¹ This research was supported in part by the National Science Foundation Grant No. ECS 8406407 and by a grant from the Ford Motor Company, Dearborn, Michigan.

Received June 1984; Revised March 1985

1. Introduction

A flexible manufacturing system consists of a set of computer numerically controlled machine tools and auxiliary equipment (such as inspection, washing, and queuing stations) that are all connected by automated material handling. The main applications to date have been metal-cutting operations, having mid-volume and batch production. Incoming castings for parts, either prismatic or rotational, are first fixtured and palletized, and then input onto the material handling system to be routed through the FMS for various machining, refixturing, and inspection operations. One or more computers, interconnected via some hierarchical control structure, coordinate most real-time functions. These include downloading the appropriate part programs to control the metal-cutting and metal-removing operations of the machine tools, as well as controlling part movements and cutting tool interchanges. Examples and descriptions of existing flexible systems can be found in Stecke (1977), Stecke and Solberg (1981b), Cavaillé et al. (1981), Barash (1982), and Stecke and Browne (1985).

Each machine tool has a limited-capacity tool magazine that stores all of the cutting tools required to perform all operations that might be performed at that machine tool. The very quick, automated tool interchange capability implies that there is no need to consider a machine set-up time between consecutive operations to be performed at the same machine tool. In addition, a part cannot be routed to a particular machine tool unless it is one of the machine types capable of performing that part's next operation and all cutting tools necessary to perform that next operation have previously been loaded into the machine tool's tool magazine.

These technological and computer requirements indicate a natural separation of the FMS management functions concerned with running the system into two distinct, but related, activities: first, the system has to be 'set up' (cutting tools loaded, ...) before production can start; then, the real-time, on-line control of part input, movement, and machining can occur. This paper is concerned mainly with the first activity, that of setting up an FMS in a good way, to allow efficient manufacturing and to utilize the system's capacity.

Managing an FMS can be more complicated

than the management of a conventional manufacturing system because there are many more manufacturing and routing options and other factors to consider. The machine tools are quite versatile and several part types can be machined simultaneously. In addition, each individual part, even those of the same part type, can have several routes through the system. Parts do not have to be machined in batches. At the same time, the machining process could and should be nearly as efficient as a well-balanced, automated transfer line.

In its entirety, the FMS set-up problem is intractable. However, with efficient and flexible production in mind, the following FMS production planning problems have been defined (Stecke (1983a)):

1. Part type selection:

From a set of part types, each having production requirements, determine a subset for immediate and simultaneous processing.

2. Machine grouping:

Partition the machine tools of each machine type into machine groups such that each machine in a particular group is able to perform the same set of operations.

3. Production ratios:

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

4. Resource allocation:

Allocate the limited numbers of pallets and fixtures of each fixture type among the selected part types.

5. Loading:

Allocate the operations and associated cutting tools of the selected set of part types among the machine groups subject to the technological and capacity constraints of the FMS.

Unlike the very infrequent set-up of a transfer line (usually only during the design and layout of the line), the above planning decisions can be made often (say every one to three weeks). Reconfiguring the machines and reloading the magazines might occur, for example, whenever the production requirements are complete for some part type. (This would free up space in a tool magazine for other cutting tools.) Reloading is also required when a rush order arrives that needs to be expedited or sometimes when a part type is to begin production. (Space may need to be found in a tool

magazine somehow, perhaps by reallocating operations and reshuffling tool assignments.) Cutters might also be changed around if the part type mix is changed and sometimes when a breakdown occurs.

To begin the planning process, production requirements are given for several of the part types that are produced on the FMS. These requirements are determined either from customer orders or forecasted demand. The usual case is that all required part types cannot be produced simultaneously on the system. There is not enough space in the tool magazines to hold all required cutting tools. Hence the part types have to be partitioned into batches, each of which is machined one at a time. Then the first planning problem (1) is to select the part types to be produced in the next batch, over some time period. Whitney and Gaul (1984) suggest a heuristic approach to select the part types for each batch. When one batch of selected part types is complete, the next begins.

One method of determining the relative production ratios (problems (3) and (4)) at which the part types should be on the system can use the mean value analysis (MVA) technique which analyzes a queueing network model. See Cavaillé and Dubois (1982) and Suri and Hildebrant (1984). A multiple class model is used to provide the steady state expected production for each of several part types. A complimentary model to use to help answer the same questions ((3) and (4)) is a Petri net. See Dubois and Stecke (1983). A Petri net is a deterministic model that can also analyze some transient and steady state effects. Both models account for the congestion as parts compete for the same limited resources (i.e., machine tools). MVA assumes random operation times while a Petri net requires deterministic processing times. Use of both models would provide bounds on the requirements specifications to operate the FMS efficiently.

The remainder of this paper focuses on a different two of these planning problems, the machine grouping problem (2) and the loading problem (5), at several levels of detail. The grouping problem is a different sort of problem because in the past, even for non-automated systems, each operation was usually assigned to one and only one machine tool. However, pooling machines into groups allows alternative routes for parts and can increase all system performance measures. The approach to solve the loading problem was usually to attempt

to achieve the best balance of workload (operations to be performed) among the machine tools.

However, an FMS is an integrated system of very versatile CNC machine tools. It seemed desirable to devise new loading and scheduling policies that could utilize the available flexibilities to the system's advantage so that machine utilization, production, and overall efficiency could increase. With these goals in mind, various FMS loading objectives and control strategies were developed and applied to data from an existing FMS, with interesting results. As measured by the production rate and machine utilizations, several objectives performed much better than that of balancing the workload, despite the fact that they produced highly unbalanced systems (Stecke (1977), Stecke and Solberg (1981b)).

Such results encouraged subsequent research by the author and others (for example, see Kusiak (1983)) to investigate FMS grouping and loading problems. Various approaches, models, results, and solutions are coordinated in this paper. In the following section, the hierarchy of problems, models, and solutions is presented. First, aggregate grouping and loading problems are defined and some previous results are reviewed. Then, similar information is provided for detailed grouping and loading problems. The interface between theoretical results and solutions at the aggregate level and the detailed level models is also given in Section 2. The use of the aggregate level model and results and the detailed level models to solve realistic grouping and loading problems is described in Section 3. Suggestions for future research and further refinements are provided in Section 4.

2. The hierarchy

2.1. Aggregate and detailed level models

At an aggregate level, a single class closed queueing network (CQN) model of multiple server queues (machines) is used to represent an FMS. This model requires only aggregate input requirements to output average values (mean production rate and mean queue lengths, for example).

This model is useful for several reasons. First, it considers congestion resulting when different parts are competing for the same machine tool. Second, CQN models have been shown to be robust, even

when they require distributional assumptions and queue disciplines quite different from the actual system distributions and queue disciplines. (See, for example, Giammo (1976), Lipsky and Church (1977), Rose (1976), Solberg (1977), and Suri (1983).) The observed robustness of steady-state results is surprising also because the model requires that most information be aggregated into average processing times and relative visit frequencies of the parts for each machine tool (or machine group). Within the framework of a CQN, the measure used here to evaluate solutions to the grouping and loading problems is the expected production rate.

Then at the detailed level, the *FMS grouping and loading problems* are formulated in all detail as nonlinear mixed integer programs. This model is useful for several reasons. First, it allows consideration of all available FMS information, such as the actual processing time, cutting tools required, and tool slots required for each operation, while considering all relevant constraints. Second, it can result in an optimum solution. Third, heuristic approaches can be developed that are based on the MIPs. The measure used to evaluate solutions is the production rate.

At each level, suppose that it has already been decided which part types (and also their relative production ratios) shall be machined simultaneously over the next production period. Pallets and fixtures of each fixture type have also been allocated among the part types to result in revised relative production ratios.

2.2. Aggregate problems and results

The information that is suppressed at the aggregate level includes the actual deterministic operation times, cutting tools and tool slots required, and machine types required. It is assumed at the aggregate level using the CQN model, without loss of generality for the purpose here, that there is a single part type and a single machine type. Grouping and loading are performed within each machine type. The generalizations will be described. The aggregate grouping and loading problems are now defined.

Aggregate grouping problem. Partition the m machine tools into g machine groups to maximize expected production.

Machine tools in the same machine group are said to be pooled, are identically tooled, and are therefore physically capable of performing the same set of operations during real-time control. Pooling, in conjunction with real-time, on-line control, implies that a part whose next operation requires one of these pooled machine tools need find only one machine free. As a result, many regular system performance measures can only improve, relative to a system with no pooling. Also, pooling machines provides redundancy in the case of machine breakdowns.

Aggregate loading problem. Allocate a total, fixed amount of work among a system of (grouped) machines to maximize expected production.

The workload assigned by parts to each machine group i is proportional to the relative frequency of visits by parts to group i multiplied by the average processing time of an operation performed by a machine tool in group i . These parameters are very easy to calculate. Then the performance measure using the CQN model, the expected production, $\text{Pr}(g, n; S, X)$, is defined as a function of g (the number of groups); n (the number of parts or pallets); S (a partition of a m machine tools into g groups) $= [s_1, s_2, \dots, s_g]$, where s_i is the number of machine tools in group i ; and X (an allocation of work among the machine groups) $= [x_1, x_2, \dots, x_g]$, where x_i is the workload assigned to group i . (See Stecke and Solberg (1981a) for additional details.) The optimal S (the best grouping) and the optimal X (the best loading) are determined.

Some previous mathematical programming results are used in conjunction with the CQN model to solve these aggregate problems. First, symmetric functions (Berge (1963)) and symmetric mathematical programming (Greenberg and Pierskalla (1970)) are applied to obtain some theoretical results concerning the optimality of balancing workloads (see Stecke and Morin (1985)). Then, CAN-Q (Solberg (1980)), a computer program that uses an efficient algorithm of Buzen (1971) to solve product-form CQNs to obtain steady-state measures, is used to provide some theoretical and operational results. The aggregate solutions that can be used (Stecke and Solberg (1985)) are now summarized.

Aggregate grouping solution. All possible partitions of m machine tools into g groups are ordered

according to expected production. In particular,

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

(ii) If constraints define that g groups of m machine tools are required, then the more unbalanced configuration provides the larger maximum expected production.

The mathematical statements of these results are now provided. The machine groups are ordered according to increasing size, that is, $s_1 \leq s_2 \leq \dots \leq s_g$. Define:

$$\underset{X}{\text{maximum}} \Pr(g, n; [s_1, \dots, s_g], X)$$

to be the maximum expected production from the particular configured system, where $X = [x_1, \dots, x_g]$ and $S = [s_1, \dots, s_g]$.

Then for any integers, i and $k > 0$, we have that:

$$(1) \underset{X}{\text{maximum}} \Pr(g, n; [s_1, \dots, s_i + k, \dots, s_g - k], X)$$

is strictly less than

$$\underset{X}{\text{maximum}} \Pr(g, n; [s_1, \dots, s_i, \dots, s_g], X);$$

$$(2) \Pr(g, n; [s_1, \dots, s_i + k, \dots, s_g - k], S)$$

is strictly greater than

$$\Pr(g, n; [s_1, \dots, s_g], S);$$

$$(3) \underset{X}{\text{maximum}} \Pr(g, n; [s_1, \dots, s_j, \dots, s_g], X)$$

is strictly greater than

$$\underset{X}{\text{maximum}} \Pr(g, n; [s_1, \dots, s_i + k, \dots, s'_j, \dots, s_g], X),$$

for $s_i + k = s_{i+1} + k = \dots = s'_j$.

Aggregate loading solution. The solution is a set of optimal allocation ratios, $x_1^*, x_2^*, \dots, x_g^*$, or the theoretically best ratios at which the g groups should be assigned work so as to maximize the expected production of the FMS. In particular,

1. If all groups contain the same number of machine tools, then a balanced workload per machine is optimal.

2. For a system of unequally sized machine groups (the better configuration), unbalancing the

assigned workload per machine is optimal. In addition, if a perfect 'unbalance' is not possible, then it is better to overload the larger group than otherwise.

2.3. Detailed problems and solutions

The detailed problems consider the actual FMS factors and constraints such as tool magazine capacity, integrality of the decision variables, processing times, and machine type requirements. The cutting tools and tool slots required for each operations need to be known in order to use the additional available magazine storage capacity that is obtained from avoiding the unnecessary duplication of cutting tool assignments as well as the tool slot overlap of some larger cutting tools. The detailed problems are now defined.

Detailed grouping problem. Partition the m machines into g groups subject to technological and capacity constraints of the FMS so as to maximize expected production.

Detailed loading problem. Allocate the operations and associated cutting tools of a selected set of part types among the machine groups subject to the technological and capacity constraints of the FMS and according to some loading objective.

These problems are formulated in all detail as nonlinear mixed integer programs (Stecke (1983a)). The solution procedures are now outlined in brief.

Detailed grouping solution. First, a nonlinear mixed integer problem can be solved to find the minimum number of machine tools of each machine type i , g_i , that are required to perform all operations of the chosen set of part types. Then the aggregate grouping solution is used to determine immediately the partition of the m machine tools into g_i groups that theoretically maximizes expected production subject to the FMS constraints. The solution to the nonlinear MIP also provides a feasible loading of operations and tools on each machine tool.

Detailed loading solution. The solution is an assignment of the operations and associated tools to machine groups. Table 1 suggests six possible loading objectives, each applicable to different

Table 1
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 number of operation assignments.
-

types of flexible manufacturing systems. The applicability of each depends on the particular manufacturing situation and problem specifics.

Each objective in Table 1 defines a nonlinear integer problem that has been mathematically formulated. Situations in which each might be applicable are described in Section 3.

Several methods have been proposed to solve the nonlinear integer problems. First, since the nonlinear terms are products of 0–1 integer variables, one method suggested to solve these problems exactly involves manually linearizing the nonlinear terms. Several linearization methods were examined and one was chosen to apply to data from an existing FMS. Several problems were solved in reasonable time (minutes) using a standard MIP code. The solutions demonstrate the value of considering tool duplication and tool slot overlap to allow additional magazine storage capacity despite the consequential increase in the size of the linear integer programs (Stecke (1983a)). The need was recognized for a specialized, self-contained, more efficient, computerized algorithm to quickly retool an FMS as often as was required.

Subsequently, an efficient branch and bound approach was suggested for a subset of the loading objectives of Table 1, in particular, the balancing objectives, with or without pooling (Berrada and Stecke (1983)). The branch and bound algorithm can solve most *FMS loading problems* in a reasonable amount of CPU time. However, for large loading problems, heuristics should be used to find good solutions. Whitney and Gaul (1984) provide a heuristic approach for a balancing loading objective, while Stecke and Talbot (1983) suggest heuristics for several of the loading objectives.

3. Implementation

Some the queuing network results can be applied directly to the detailed problems. Otherwise, an aggregate level CQN problem is solved to provide some of the input for the detailed formulations. Since the grouping problem naturally precedes the loading problem in practice (operations are assigned to groups of machines), it is solved first in the suggested hierarchical procedure. It may be possible to solve the grouping and loading problems together. Alternatively, iteration between the two problems may be required to converge to good working solutions.

Recall that for the part types under consideration for simultaneous machining, the machine type, tools and tool sizes, and processing time required for each operation are known.

3.1. Realistic grouping problems

The procedure to follow to solve real *FMS grouping problems* is now provided.

Step 1. For each machine type, determine both the total operation time requirement and an upper bound on the total number of tool slots required. The latter is found by summing the number of tool slots required by all cutting tools for all operations that require that machine type. This number is an upper bound because neither tool duplication nor tool overlap is accounted for yet. Dividing the latter by the number of slots in each tool magazine, and rounding the fraction up, gives an upper bound on the number of machine tools of each type needed to perform all required operations.

(a) If each *upper bound* on the number of required machines of each machine type equals the *minimum* number of required machines of each machine type (g_i), go to Step 2. (For each machine type, the factors necessary to determine whether the upper bound is minimum include: the number of slots in the remainder of the above quotient, the amount of cutting tool duplication, and even the number of large tools.)

(b) For any upper bound that might be larger than the minimum number of machines for a machine type, say i , the nonlinear formulation for grouping (see Stecke (1983a)) can be applied to find the minimum number of required machines (g_i).

Step 2. The aggregate grouping solution immediately provides the optimal partition of the m_i machine tools of type i into g_i groups that maximizes expected production.

An additional, useful output from the first step is a candidate loading, which is a feasible assignment of the operations and cutting tools among the $\sum g_i$ machine groups that conforms to the maximal pooling constraints that are also found.

3.2. Realistic loading problems

The aggregate loading solutions, obtained through the use of the CQN model, relate only to the first, third, and fourth objectives of Table 1, the balancing or unbalancing objectives. These three objectives are discussed shortly. We also describe how the aggregate results can be used and we describe procedures to use to provide solutions at the aggregate level; both serve as input to the detailed formulations dictated by these three objectives. The nonlinear integer problems derived from objectives 2, 5, and 6 are also discussed. First, various situations in which each loading objective is applicable are described.

3.2.1. Applicability of each loading objective

The aggregate grouping solution indicates that it is best to pool as much as possible. In addition, if maximum pooling is not technologically feasible, the best partition of m machines into g groups is known. However, if many different cutting tools are needed to manufacture the current set of operations, then perhaps no pooling is possible. At the same time, there may be enough spare storage capacity in some tool magazines to allow several operations to be assigned to more than one machine tool. We call such multiple operation assignments: *partial pooling*. The motivation for the fifth and sixth objectives is that via some partial pooling, they allow an increase in system flexibility (even when no pooling of machines is possible) by providing alternative routes for at least some of the parts through the FMS.

The second objective, to minimize movements, is very different from the others. The idea here is for a particular part to remain at its current machine as long as that machine tool can perform the next consecutive operation(s), rather than to move it for the purpose of better balancing the machine

workloads. By remaining on the same machine, unnecessary travel time (to a next required machine) and waiting time (for the potentially busy next machine to become free) are saved. This objective can be seen to be applicable when travel time is long relative to processing time and also when the material handling system is a bottleneck. In addition, it has previously been demonstrated that minimizing movements proved better than balancing when both objectives were applied using data from an existing FMS. In this case, the transporters did not slow down the operation, travel time was short relative to operation time, and often travel could occur while the next machine was completing its current operation. Minimizing movements was shown to be superior even though the resultant system was highly unbalanced and there was no pooling (Stecke (1977)). In general, additional system performance benefits can also be realized when the objective of minimizing movements is applied in conjunction with pooling objectives.

3.2.2. Relationship between the aggregate and detailed loading problems

There is a relationship between the two aggregate and detailed levels for both the balancing objectives, 1 and 3, and the unbalancing objective, 4. If the aggregate grouping solution is a balanced configuration of groups, then a balanced loading per machine is also optimal. Then the loading problem is solved by using the nonlinear formulation associated with objective 1 or 3. This nonlinear integer formulation for balancing can be solved either by linearizing (see Stecke (1983a)) or by branch and bound (see Berrada and Stecke (1983)) for optimal solution methods. For very large problems, heuristics can be used (see Stecke and Talbot (1983) and Whitney and Gaul (1984)).

If the aggregate grouping solution is an unbalanced configuration, then the optimal allocation ratios, x_i^* , must be determined before the unbalancing loading objective 4 can be applied. The determination of the optimal allocation ratios is now described. To simplify notation, it can be assumed without loss of generality that there is only one machine type, since the algorithm is applied identically for each machine type. Similarly, it is assumed without loss of generality that there are n identical parts or pallets in the system. If n has not yet been determined, the algorithm

can be applied once, for a specified, relevant range of n , to also help determine the appropriate number of pallets that should be in the system. In general, the problem is to determine the minimum number of pallets (in-process inventory) required to maintain desired production ratios. See Shanthikumar and Stecke (1986).

In the following, since the configuration of machine tools into groups is unbalanced, $s_i \neq s_j$, for some i and j .

Algorithm to determine the optimal allocation ratios

1. Order the g machine groups so that $s_1 \leq s_2 \leq \dots \leq s_g$.
2. A balanced allocation is $X = [s_1, s_2, \dots, s_g]$. For each i , the 'direction' (either less than, greater than, or equal to s_i) of each optimal allocation ratio, x_i^* , is known (Stecke and Solberg (1985)). Then, if the middle machine tool ($m/2$) is in group i , a reasonable initial starting point in the search for the optimal ratios is, for some $\epsilon > 0$,

$$[x_1, \dots, x_g] = [s_1 - (1 - i)\epsilon, \dots, s_{i-1} - \epsilon, s_i, s_{i+1} + \epsilon, s_{i+2} + 2\epsilon, \dots, s_g + (g - i)\epsilon].$$

3. Fix $g - 2$ of the x_k . Vary x_i and x_j , $i < j$. The production function, $\text{Pr}(g, n; S, X)$, is evaluated along the hyperplane

$$x_i = \sum_{\substack{k=1 \\ k \neq i, j}}^g x_k - x_j = K - x_j,$$

for a range of x_i and x_j , such that $x_i/s_i \in [a, 1]$ and $x_j/s_j \in [k - a, k - 1]$, for some fixed $a < 1$ (i.e., $a = 0.7$). The maximum expected production on the hyperplane is found. Many production functions can be evaluated in seconds of CPU time by using a computer program that is a variation of Solberg's CAN-Q (1980). (A listing of the adapted program can be found in Stecke and Solberg (1981a), which also provides several examples that demonstrate the use of the program.)

4. The optimal allocation ratios that produce the maximum expected production have been found for the particular hyperplane. The computer program also provides sensitivity information on the change in the expected production that would result from a change in a workload parameter.

This indicates both a direction and magnitude of a possible change to use in the search for the optimal x_i 's. This sensitivity information is used to choose a different $g - 2$ of the x_k to fix. Return to Step 3. Continue to search along different hyperplanes until the sensitivity analysis indicates that the optimum allocation ratios, x_i^* , have been determined.

Since the production function, $\text{Pr}(g, n; S, X)$, monotonically increases and then decreases as a function of X , it is possible to develop a computer program to search for the unique, optimal allocation ratios. See Stecke and Solberg (1981a, 1985) and Stecke (1983b) for information on such properties of the production function.

3.2.3. Aggregate results as input for the detailed loading models

When the optimal allocation ratios, x_i^* , are determined, they are used in the nonlinear objective function formulated from the fourth loading objective, unbalancing. Each loading objective in Table 1 gives rise to a different nonlinear integer program. The mathematical formulations, given in Stecke (1983a), can be solved either using an available mixed integer code (after linearizing) or directly, which yields the optimal allocation of operations and cutting tools among the machine groups of the flexible manufacturing system. If the appropriate loading objective is balancing, then the branch and bound algorithm (Berrada and Stecke (1983)) can be used.

4. Future research

At present, the Algorithm to find the optimal allocation ratios, which is described in Section 3.2.2, is implemented by first running the computer program and then using the output to determine which variables to fix and vary subsequently. However, it would be relatively straightforward to close the loop and to automatically find these optimal ratios, by imbedding an appropriate search strategy into the closed queueing network program.

The solution methodologies reported here are mostly optimum-producing. The nonlinear integer problems have been solved for real grouping and loading problems, defined from real data, on a

large computer system (CDC 6600). A standard mixed integer code was used. However, real-time FMS control requires very quick solutions to these planning problems. Much companies do not have a mixed integer code available. Also, these problems would be solved often enough to warrant a special code that is self contained. Such a code does exist for a balancing loading objective, and is needed for the other loading objectives.

Future research needs include the development of efficient heuristic algorithms that provide good solutions to FMS grouping and loading problems. This is especially true if larger flexible system are to be developed having more part numbers introduced. Some heuristics development has taken place, mostly for the balancing objective of the FMS loading problem. All of these procedures should be executable on minicomputers that are now available in essentially real-time.

Additional research is also required to determine efficient means to solve *all* of the planning problems. Much of the software that exists to run these highly automated systems is not very good and is not very flexible.

References

- Barash, Moshe M. (1982), "Computerized manufacturing systems for discrete products", in: G. Salvendy (ed.), *The Handbook of Industrial Engineering*, Wiley, New York.
- Berge, Claude (1963), *Topological Spaces*, Macmillan, New York.
- Berrada, Mohammed, and Stecke, Kathryn E. (1983), "A branch and bound approach for machine loading in flexible manufacturing systems", Working paper no. 329, Graduate School of Business Administration, The University of Michigan, Ann Arbor, MI.
- Buzen, Jeffrey P. (1971), "Queueing network models of multi-programming", Ph.D. dissertation, Harvard University, Cambridge, MA.
- Cavallé, Jean-Bernard, and Dubois, Didier (1982), "Heuristic methods based on mean-value analysis for flexible manufacturing systems performance evaluation", in: *Proceedings of the 21st IEEE Conference on Decision and Control*, Orlando, FL, pp. 1061-1065.
- Cavallé, Jean-Bernard, Forestier, J.P., and Bel, Gérard (1981), "A simulation program for analysis and design of a flexible manufacturing system", in: *Proceedings of the IEEE Conference on Cybernetics and Society*, Atlanta, GA, pp. 257-259.
- Dubois, Didier, and Stecke, Kathryn E. (1983), "Using Petri nets to represent production processes", in: *Proceedings of the 22nd IEEE Conference on Decision and Control*, San Antonio, TX.
- Giammo, T. (1976), "Validation of a computer performance model of the exponential queueing network family", *Acta Informatica* 17 (2), 137-152.
- Greenberg, Harvey J., and Pierskalla, William P. (1970), "Symmetric mathematical programs", *Management Science* 16 (5), 309-312.
- Kusiak, Andrew (1983), "Loading models in flexible manufacturing systems", Working paper no. 05/83, Department of Industrial Engineering, Technical University of Nova Scotia, Halifax, Nova Scotia, Canada.
- Lipsky, L., and Church, J.D. (1977), "Applications of a queueing network model for a computer system", *Computing Surveys* 9, 205-221.
- Rose, C.A. (1976), "Validation of a queueing model with classes of customers", in: *Proceedings of the International Symposium on Computer Performance Modeling, Measurement, and Evaluation*, Harvard University, Cambridge, MA, pp. 318-325.
- Shanthikumar, J. George, and Stecke, Kathryn E. (1986), "Reducing work-in-process inventory in certain classes of flexible manufacturing systems", *European Journal of Operational Research*, forthcoming.
- Solberg, James J. (1977), "A mathematical model of computerized manufacturing systems", in: *Proceedings of the 4th International Conference on Production Research*, Tokyo, Japan.
- Solberg, James J. (1980), "CAN-Q user's guide", Report no. 9 (revised), NSF GRANT No. APR74 15256, School of Industrial Engineering, Purdue University, W. Lafayette, IN.
- Stecke, Kathryn E. (1977), "Experimental investigation of a computerized manufacturing system", Master's thesis, Purdue University, West Lafayette, IN.
- Stecke, Kathryn E. (1983a), "Formulation and solution of nonlinear integer production planning problems for flexible manufacturing systems", *Management Science* 29 (3), 273-288.
- Stecke, Kathryn E. (1983b), "On the nonconcavity of throughput in certain closed queueing networks", Working paper no. 356, Graduate School of Business Administration, The University of Michigan, Ann Arbor, MI.
- Stecke, Kathryn E., and Browne, Jim, (1985), "Variations in flexible manufacturing systems according to the relevant types of automated materials handling", *Material Flow* 2 (2), 179-185.
- Stecke, Kathryn E., and Morin, Thomas L. (1985), "Optimality of balancing workloads in certain types of flexible manufacturing systems", *European Journal of Operational Research* 20 (1), 68-82.
- Stecke, Kathryn E., and Solberg, James J. (1981a), "The CMS loading problem", Report no. 20, NSF Grant no. APR 74 15256, School of Industrial Engineering, Purdue University, West Lafayette, IN.
- Stecke, Kathryn E., and Solberg, James J. (1981b), "Loading and control policies for a flexible manufacturing system", *International Journal of Production Research* 19 (5), 481-490.
- Stecke, Kathryn E., and Solberg, James J. (1985), "The optimality of unbalancing both workloads and machine group sizes in closed queueing networks of multiserver queues", *Operations Research* 33 (4), 822-910.
- Stecke, Kathryn E., and Talbot, F. Brian (1983), "Heuristic loading algorithms for flexible manufacturing systems", in:

- Proceedings of the Seventh International Conference on Production Research*, Windsor, Ontario, Canada, August 22–24.
- Suri, Rajan (1983), “Robustness of queueing network formulas”, *Journal of the Association for Computing Machinery* 30 (3), 564–594.
- Suri, Rajan, and Hildebrant, Richard R. (1984), “Modeling flexible manufacturing systems using mean-value analysis”, *Journal of Manufacturing Systems* 3 (1), 27–38.
- Whitney, Cynthia K., and Gaul, Thomas S. (1984), “Sequential decision procedures for batching and balancing in FMSs”, in: *Proceedings of the First ORSA/TIMS Special Interest Conference on Flexible Manufacturing Systems: Operations Research Models and Applications*, Ann Arbor, MI, August 15–17, pp. 243–248.