The International Journal of Flexible Manufacturing Systems, 3 (1991): 213-230
© 1991 Kluwer Academic Publishers, Boston. Manufactured in The Netherlands.

The Optimal Configuration and Workload Allocation Problem in Flexible Manufacturing Systems

HEUNGSOON FELIX LEE
Department of Industrial Engineering, Southern Illinois University, Edwardsville, IL 62026-1802

MANDYAM M. SRINIVASAN
Department of Industrial and Operations Engineering, The University of Michigan, Ann Arbor, MI 48109-2117

CANDACE ARAI YANO
Department of Industrial and Operations Engineering, The University of Michigan, Ann Arbor, MI 48109-2117

Abstract. In this article we consider the problem of determining the minimum cost configuration (number of machines and pallets) for a flexible manufacturing system with the constraint of meeting a prespecified throughput, while *simultaneously* allocating the total workload among the machines (or groups of machines). Our procedure allows consideration of upper and lower bounds on the workload at each machine group. These bounds arise as a consequence of precedence constraints among the various operations and/or limitations on the number or combinations of operations that can be assigned to a machine because of constraints on tool slots or the space required to store assembly components. Earlier work on problems of this nature assumes that the workload allocation is given. For the single-machine-type problem we develop an efficient implicit enumeration procedure that uses fathoming rules to eliminate dominated configurations, and we present computational results. We discuss how this procedure can be used as a building block in solving the problem with multiple machine types.

Key Words: optimal configuration, workload allocation problem, precedence constraints, closed queueing networks

1. Introduction

A flexible manufacturing system (FMS) is a computer-controlled system of numerically controlled machines and automated material-handling equipment. Each machine is capable of performing a variety of operations with minimal changeover times between operations. When a system is being designed, one critical decision is the number of machines of each type to be purchased. Another important decision is the number of jobs (or pallets of jobs) circulating in the system, since this has a signficant impact on the number and type of material-handling equipment required. These decision variables define a system configuration, and normally such a configuration is maintained for several years.

The appropriate configuration choice, however, is influenced by the allocation of workload among the various machines (or machine groups). Of course, the total workload of each machine type may change over time as the product mix changes, and the "true" optimal configuration should consider these dynamics. We consider a static (steady-state) situation here, and generalize earlier work by taking advantage of the ability to allocate workload among machines of the same type while simultaneously considering other practical constraints on the workload allocation.

Our procedure can be used in several different ways. First, it can be used to identify a set of viable, cost-effective system configurations by applying it to a variety of realistic workloads and related workload allocation constraints. The robustness of these configurations to product mix changes could then be evaluated. Second, if it is difficult to predict changes in product mix, the procedure can be applied to the new mix to determine how the system configuration and workload allocation should be modified. Third, even if the number of machines of each type and the number of jobs is fixed, our procedure can be used to regroup the machines of each type and reallocate the workload among them as the product mix changes.

We investigate the problem of simultaneously determining the minimum cost configuration and the corresponding optimal workload allocation subject to a constraint on system throughput for an FMS that consists of a given number of stations. Each station may have one or more identical machines of a specified type. We assume that the total workload to be allocated among stations of the same type is given. The decisions to be made are the number of machines (servers) and the workload allocated to each station, and the number of pallets.

If there were no constraints on the allocation of the workload among the machines, the optimal solution would have a single station for each machine type, with all of the workload for that machine type assigned to it. Our procedure allows consideration of upper and lower bounds on the workload at each station. The upper bounds arise as a consequence of precedence constraints among the various operations and/or limitations on the number or combinations of operations that can be assigned to a station. The latter may be a result of constraints on tool slots or the space to store assembly components nearby (e.g., within the reach of assembly robots). Nontrivial lower bounds may arise as a consequence of upper bounds at other stations, as we demonstrate through an example in appendix 1. Earlier work on problems of this type assumes that the workload allocation is given.

In order to solve this problem, we must evaluate several candidate configurations to ascertain whether the throughput constraint is satisfied. To accomplish this, we use the algorithm of Lee, Srinivasan, and Yano (LSY) (1991) to solve the subproblem of allocating the total workload among the stations to maximize system throughput within the constraints imposed by the upper and lower bounds on the workload at each station. In order to reduce the number of candidate configurations that must be evaluated, we develop fathoming rules to eliminate dominated configurations. We develop a procedure to obtain the optimal configuration and workload allocation for the case where all the machines are of the same type. We also discuss how this procedure can be used as a building block to solve the problem with multiple machine types.

Previous research on related problems has typically used closed queueing networks (CQNs) with the assumptions of exponential service, first-come-first-served service discipline, and arbitrary routings allowing multiple visits to a station to represent FMSs. Under these assumptions, the throughput function has the well-known product form (Gordon and Newell, 1967) and is therefore relatively easy to compute. The popularity of using queueing networks to model FMSs stems from the ability of these networks to capture the effects of congestion on throughput and queue lengths. Our model also assumes that the FMS is represented by a product form CQN (see Solberg (1977) and Suri and Hildebrant (1984) for CQN modeling of FMSs and support for its applicability).

Vinod and Solberg (1985) and Dallery and Frein (1986) study the problem of finding the configuration that satisfies throughput requirements at minimum cost. Both capital equipment and operating costs are considered in their objectives. They assume that the number of stations and the workloads for each are given. (A station is represented by a multiple-server node in the queueing network.) The decisions are the number of machines to assign to each station, the number of pallets, and the number of automated guided vehicles (AGVs) where applicable.

Yao and Shanthikumar (1986) and Shanthikumar and Yao (1987, 1988) study the problem of allocating servers to stations to maximize throughput. They assume that the number of stations and pallets and the workload at each station are known. Their results suggest that a greedy allocation procedure is optimal.

Various algorithms are available to solve the workload allocation in a product form CQN under the assumption that the number of stations, the number of machines at each station, the number of pallets, and the number of AGVs are given (see, for example, Yao, 1985; Stecke and Solberg, 1985; Stecke, 1986; Lee, Srinivasan, and Yano, 1991). In our experiments, we will use the LSY algorithm. These authors derive several properties of the optimal solution under the assumption that the throughput is a pseudo-concave function of the workload allocation. Under this assumption, the first-order conditions are both necessary and sufficient for optimality. These properties are then used as the basis for an efficient reduced gradient algorithm to find the optimal workload allocation when there are lower and upper bounds on the workload for each station.

Previous research has thus considered obtaining either the minimum cost configuration assuming a given workload allocation, or the optimal (unconstrained) allocation of the total workload assuming a given system configuration. Our work differs in that we consider obtaining the optimal system configuration and the optimal workload allocation *simultaneously.*

We assume that each operation can be done by only one type of machine, and that the machine types have been predetermined. Hence, for each machine type there is an aggregate workload that is the total processing time for all the operations that can be performed by that machine type. We assume that the total cost of machines of a given type depends only upon the total number of machines of that type, and is independent of where the machines are physically located or what operations each machine actually performs.

The remainder of the article is divided as follows. In section 2, we state a mathematical formulation of the minimum cost configuration problem for a system with a single machine type. In section 3, we present an optimal algorithm for this problem, including fathoming methods to eliminate dominated configurations. Related computational results appear in section 4. In section 5, we present a solution procedure for the problem with multiple machine types. Section 5 also considers the system with batch transfer, where more than one part is transferred between stations at a time. We conclude with a brief summary and discussion in section 6.

2. Problem formulation

As mentioned earlier, the optimal configuration and workload allocation problem is to simultaneously determine the optimal workload allocation and the number of machines (servers) and pallets required to achieve a prespecified throughput at minimum cost. Here we assume that the number of stations, M, is given. However, if the number of stations is also a decision variable, one can simply solve this problem for several values of M. The total workload, TW, is the total expected machine processing time for one part. The workload allocation is specified by the vector $\bar{\mathbf{W}} = (W_0, W_1, \ldots, W_M)$ where W_0 is the total expected material-handling time for one part, which is assumed to be a known constant, and W_i denotes the workload at station i, $i = 1, \ldots, M$. The number of servers at station i, S_i, $i = 1, \ldots, M$, and the number of pallets, N, define a configuration. For ease of presentation, we define $\bar{\mathbf{S}} = (S_1, \ldots, S_M)$. We initially assume each pallet carries only one part, which is common when parts are relatively large, but later relax this assumption.

We first consider a simple FMS where there is only one type of machine. We assume that the material-handling system (MHS) is a delay node. This means that there is a delay in transferring a part from one machine to another, but there is no contention for the MHS that results in any additional (queueing) delays. Typically, these systems are designed to prevent them from becoming bottlenecks, so this assumption is reasonable for many systems. Examples include loop conveyors and dedicated (stop-and-go) AGVs. Material-handling systems for which considerable contention occurs may be modeled by representing them as other (processing) stations in the system.

The cost function, $z(N, K)$, is permitted to be any function that increases with N and

$$K = \sum_{i=1}^{M} S_i.$$

Thus, the annualized cost of the machines, material-handling equipment, pallets, and work-in-process inventory can be incorporated into the objective function. A mathematical formulation of the problem is to

Pl: Minimize $z(N, K)$

subject to:

$$K = \sum_{i=1}^{M} S_i, \tag{1}$$

$$\mathrm{TH}(M, N, \bar{\mathbf{S}}, \bar{\mathbf{W}}) \geq d, \tag{2}$$

$$\sum_{i=1}^{M} W_i = \mathrm{TW}, \tag{3}$$

$$L_i \leq W_i \leq U_i, \qquad i = 1, \ldots, M, \tag{4}$$

where

$\mathrm{TH}(M, N, \bar{\mathbf{S}}, \bar{\mathbf{W}})$ = throughput of the system, given M, N, $\bar{\mathbf{S}}$, $\bar{\mathbf{W}}$,
d = throughput requirement,
L_i = lower bound on the workload at station i, and
U_i = upper bound on the workload at station i.

As mentioned earlier, the upper and lower bounds may be consequences of precedence relations among operations and/or constraints on the number or combinations of operations that can be assigned simultaneously to the station. In appendix 1, we give an example to show how the upper and lower bounds are affected by precedence relations in a flow system. It should be intuitively clear how tool slot limitations, or constraints on the combination of assembly operations induced by consideration of the space required to store components nearby, will affect the upper bounds on workloads. Deriving tight bounds is not always easy, partly because the bounds at one station may influence the bounds at other stations. In many situations, however, they may be specified on the basis of experience.

3. An optimal solution procedure

We now consider a solution procedure for determining the optimal configuration and associated workload allocation for the FMS. We first present an overview of our procedure. This is followed by a detailed description of each step of the procedure, including fathoming methods that eliminate dominated configurations.

3.1. Procedure to find the optimal configuration and workload allocation

1. Find an initial feasible configuration and workload allocation, N^{I}, $\bar{\mathbf{S}}^{\mathrm{I}}$, and $\bar{\mathbf{W}}^{\mathrm{I}}$, and set the incumbent equal to this solution. Let z represent the cost of the incumbent.
2. Find lower bounds on the number of pallets and the total number of servers, denoted as N^{LB} and K^{LB}, respectively, below which the prespecified throughput cannot be satisfied.
3. Implicitly enumerate over values of N and K satisfying $N \geq N^{\mathrm{LB}}$, $K \geq K^{\mathrm{LB}}$, and $z(N, K) < z$.

The initial feasible solution in step 1 is obtained in the following manner. First, an initial feasible workload allocation is obtained by solving the workload allocation problem under the assumption that each station has an identical number of servers. Since balancing the workloads is optimal (for the unconstrained problem) when each station has the same number of servers, for reasonable upper and lower bounds on the workloads the resulting workload allocation is nearly balanced.

Given this workload allocation, an initial feasible configuration is then obtained using the method of Dallery and Frein (1986). Since the initial feasible workload is nearly balanced, the initial feasible configuration generally has a similar number of servers at each station and, consequently, is easy to identify. This "balanced" solution serves as an initial solution for problem Pl.

The lower bounds, N^{LB} and K^{LB}, in step 2 can be derived using the asymptotic bound analysis of Muntz and Wong (1974). Using this method, N^{LB} is $\lceil d \cdot (\mathrm{TW} + W_0) \rceil$ where $\lceil x \rceil$ is the smallest integer greater than or equal to x. This simply says that the number of pallets in the system should be at least as large as the demand (arrival) rate multiplied by the minimum sojourn time in the system. L^{LB} is

$$\max \left(\lceil d \cdot \mathrm{TW} \rceil ,\ \sum_{i=1}^{M} S_i^{\mathrm{LB}} \right)$$

where S_i^{LB} is the lower bound on the number of servers at station i and is given as $S_i^{\mathrm{LB}} = [d \cdot L_i]$ with $[y]$ denoting the smallest integer greater than y. This essentially says that the total number of servers must be large enough so that the total system utilization and the utilization levels of the individual stations are less than one. Dallery and Frein (1986) also use asymptotic bound analysis to obtain lower bounds on the number of pallets and machines.

The implicit enumeration of step 3 is executed by considering all undominated combinations of N and K. For each (N, K) pair, we must determine whether there is a feasible $\bar{\mathbf{S}}$ and $\bar{\mathbf{W}}$ for P1. Related to this decision problem, we define another problem, which is formulated as follows:

P2:

Maximize $\mathrm{TH}(\bar{\mathbf{S}}, \bar{\mathbf{W}})$

subject to constraints (1), (3), (4) of P1.

Denote as $\bar{\mathbf{S}}^*(N, K)$ and $\bar{\mathbf{W}}^*(N, K)$ the optimum solution to P2. Clearly, when $\mathrm{TH}(\bar{\mathbf{S}}^*(N, K), \bar{\mathbf{W}}^*(N, K)) < d$, there is no feasible solution to the problem for the given N and K. Later in this section we provide a procedure to solve P2. Before doing so, we present two lemmas that permit us to eliminate some dominated (N, K) pairs. Proofs of the lemmas appear in appendix 2.

Lemma 1. If $\mathrm{TH}(\bar{\mathbf{S}}^*(N, K), \bar{\mathbf{W}}^*(N, K)) < d$, then $\mathrm{TH}\,(\bar{\mathbf{S}}^*(N, K-1), \bar{\mathbf{W}}^*(N, K-1)) < d$.

Lemma 2. If $\mathrm{TH}(\bar{\mathbf{S}}^*(N, K), \bar{\mathbf{W}}^*(N, K)) < d$, then $\mathrm{TH}(\bar{\mathbf{S}}^*(N-1, K), \bar{\mathbf{W}}^*(N-1, K)) < d$.

We solve P2 by generating all partitions of K machines among M stations and sequencing them from the most unbalanced to the most balanced. (This ranking turns out to be a lexicographic ordering.) Observe that for each partition, there are several different $\bar{\mathbf{S}}$'s, each of which corresponds to a different permutation of the station indices. For example, there is only one way to partition three machines into two groups: two machines in one group and one machine in the other. This partition gives two different $\bar{\mathbf{S}}$'s: (1, 2) and (2, 1). In order to distinguish between partitions and the various $\bar{\mathbf{S}}$'s, a partition is denoted by $\bar{\mathbf{G}} = (G_1, \ldots, G_M)$, where $G_1 \geq G_2 \geq \ldots G_M$.

Our rationale for sequencing the partitions from the most unbalanced to the most balanced is based upon the empirical observation by Stecke and Solberg (1985) that more unbalanced configurations achieve a higher throughput. Justification for this conjecture is based on the pooling effect (Kleinrock, 1976): a larger group of pooled machines can be loaded more heavily simply because pooled servers can achieve a higher utilization than an equal number of single servers given the same average customer waiting time.

For each candidate $\bar{S}$, we use the LSY algorithm to determine the workload allocation that maximizes throughput subject to constraints (3) and (4) of P1. In the article by LSY, the throughput for a CQN is shown to be a pseudoconcave function of the workloads in special cases. Based on the conjecture that the function is pseudoconcave in general, two algorithms are developed: a reduced-gradient procedure (Avriel, 1976), and a fixed-point procedure (Saigal, 1977). The reduced-gradient procedure is basically an ascent algorithm in which all of the variables can be changed simultaneously. The fixed-point procedure divides the entire feasible region into many small pieces called simplexes, and traverse a series of simplexes systematically until one is found which contains a point satisfying the Kuhn-Tucker (first-order) conditions. At this time, it starts to search inside the particular simplex by further dividing it into smaller simplexes and repeats the same process. The procedure terminates when the simplex size is small enough.

Both procedures take advantage of the fact that satisfaction of the Kuhn-Tucker conditions are both necessary and sufficient for optimality in a linearly constrained problem if the objective function is pseudoconcave. (Weaker forms of concavity require computation of the Hessian to find the optimal solution.) The fixed-point procedure also uses the result that if the number of pallets in the system is greater than the maximum number of servers at any station, the optimal solution is an interior Brouwer's fixed point. This fixed point can be found by the Eaves-Saigal (Saigal, 1977) procedure, which converges quadratically for unconstrained problems.

The workload allocation problem is complicated by the existence of upper and lower bounds on the workloads. Incorporating workload bounds into the fixed-point procedure is easy, but quadratic convergence is no longer guaranteed because of the manner in which constraints are handled by the procedure. In the case of the reduced-gradient procedure, it is necessary to find an initial feasible solution, and a simple algorithm to find such a solution is presented in appendix 3. We use the reduced gradient procedure in our computational experiments. A computational comparison of the reduced gradient and the Eaves-Saigal algorithms appears in Lee, Srinivasan, and Yano (1991).

Each candidate $\bar{S}$ is considered in turn until a configuration that satisfies the throughput constraint is identified or until all configurations have been considered and none satisfies the constraint. Some of the $\bar{S}$'s can be eliminated from consideration using the lemmas below. Proofs of the lemmas appear in appendix 2.

Lemma 3. If $U_i \leq L_k$ for any i and k, then we only need to consider $\bar{S}$ such that $S_i \leq S_k$.

Lemma 4. If $L_i \leq L_k \leq U_i \leq U_k$ for any i and k, then we only need to consider $\bar{S}$ such that $S_i \leq S_k$.

Qualitatively, lemmas 3 and 4 state that a station with a greater workload should be assigned a larger number of servers.

Lemmas 1 and 2 eliminate many (N, K) pairs. For each of the (N, K) pairs that still remains for consideration, we need to consider all possible partitions of K among the M stations, and each such partition will give rise to several possible permutations of the server vector. Lemmas 3 and 4 eliminate many of these permutations from consideration. In addition, other permutations can be eliminated from consideration using the following observation, which is based upon the assumption of unimodality of the throughput function (Stecke and Solberg, 1985; Stecke, 1986; Lee et al., 1991).

Remark. Let $\bar{\mathbf{W}}^1$ be the optimal workload allocation for a given server vector $\bar{\mathbf{S}}^1$. Also let I denote the set of stations for which $L_i < W_i^1 < U_i$. Suppose we now permute the server vector and its corresponding workload vector for just those stations in the set I to get a new server vector $\bar{\mathbf{S}}^2$ and a workload vector $\bar{\mathbf{W}}^2$. If, for this configuration, we have $L_i \leq W_i^2 \leq U_i$ for all i, then this workload allocation is also optimal, since the throughput of these two configurations is identical.

The above remark enables us to eliminate the server vector $\bar{\mathbf{S}}^2$ from consideration in the search process for such cases. It should also be noted that any $\bar{\mathbf{S}}$ with $S_i < S_i^{\mathrm{LB}}$ for any i can be eliminated from consideration, where S_i^{LB} is obtained from asymptotic bound analysis.

We now elaborate on the implicit enumeration over N and K, which is step 3 of the procedure given earlier. In the implicity enumeration, we use the results in lemmas 1 and 2 and the fact that $z(N, K)$ is increasing in N and K to fathom solutions. We use N^{p} and K^{p} to refer to the values of N and K, respectively, in the present incumbent solution. The initial solution, N^{I} and $\bar{\mathbf{S}}^{\mathrm{I}}$ provides the first incumbent solution. A description of the procedure follows.

3.2. *The implicit enumeration procedure for step 3*

{Step 3a finds the next incumbent solution by decreasing K as much as possible from the initial solution K^{I} obtained from step 1 while maintaining feasibility.}

(3a) For $N = N^{\mathrm{I}}$, find the smallest value of $K \geq K^{\mathrm{LB}}$ for which $\mathrm{TH}(\bar{\mathbf{S}}^*, \bar{\mathbf{W}}^*) \geq d$. This provides an incumbent solution, which we denote as $(N^{\mathrm{f}}, K^{\mathrm{f}})$. Set $(N^{\mathrm{p}}, K^{\mathrm{p}}) = (N^{\mathrm{f}}, K^{\mathrm{f}})$. {*Steps 3b and 3c search over* $N > N^{\mathrm{f}}$. For each value of N, the smallest feasible value of K is found. Whenever a feasible solution with lower cost is found, the incumbent solution is updated.}

(3b) Increase N by one and find the largest K such that $z(N, K) < z(N^{\mathrm{p}}, K^{\mathrm{p}})$.

(3c) If $K < K^{\mathrm{LB}}$, then set K to $K^{\mathrm{f}} - 1$, and go to step 3d. If $K \geq K^{\mathrm{LB}}$ and $\mathrm{TH}(\bar{\mathbf{S}}^*, \bar{\mathbf{W}}^*) < d$, then go to step 3b. In all other cases, update the incumbent solution, reduce K by one, and repeat this step. {Step 3d and 3e search over $K \geq K^{\mathrm{f}}$. For each value of K, the smallest feasible value of N is found. Whenever a feasible solution with lower cost is found, the incumbent solution is updated.}

(3d) Increase K by one and find the largest N such that $z(N, K) < z(N^{\mathrm{p}}, K^{\mathrm{p}})$.

(3e) If $N < N^{\mathrm{LB}}$, then go to step 3f. If $N \geq N^{\mathrm{LB}}$ and $\mathrm{TH}(\bar{\mathbf{S}}^*, \bar{\mathbf{W}}^*) < d$, then go to step 3d. In all other cases, update the incumbent solution, reduce N by one, and repeat this step.

(3f) The incumbent solution is optimal. Terminate.

Example. We provide an example to clarify the solution procedure for problem P1. Let $M = 3$ and $W_0 = 8$. Also let the processing capacity of a machine (i.e., total service time available from a machine during a period) be 960 time units. The number of units required during this period is 100 items, giving a throughput requirement of 100/960 items per unit time. The problem is to:

$$\text{Minimize} \quad z(N, K) = 600\,N + 5000\,K$$

subject to:

$$K = \sum_{i=1}^{3} S_i,$$

$$\text{TH}(M, N, \bar{\mathbf{S}}, \bar{\mathbf{W}}) \geq 100/960,$$

$$\sum_{i=1}^{3} W_i = 30,$$

$$5 \leq W_1 \leq 10, \quad 10 \leq W_2 \leq 15, \quad 5 \leq W_3 \leq 20.$$

From Lemmas 3 and 4, we must have $S_1 \leq S_2$ and $S_1 \leq S_3$. We execute the three steps outlined in the procedure.

1. Initial solution: $\bar{\mathbf{W}}^{\text{I}} = (10, 10, 10)$, $\bar{\mathbf{S}}^{\text{I}} = (2, 2, 2)$, $K^{\text{I}} = 6$, $N^{\text{I}} = 5$, with cost $z(5, 6) = 33{,}000$.
2. Lower bounds: $\bar{\mathbf{S}}^{\text{LB}} = (1, 2, 1)$, $K^{\text{LB}} = 4$, and $N^{\text{LB}} = 4$.
3. The implicit enumeration procedure:

(3a) With $N = N^{\text{I}} = 5$, we decrease K as much as possible, while maintaining feasibility. We obtain $K = 5$, and $z(5, 5) = 28{,}000$.

When $K = 5$, there are two possible partitions: $\bar{\mathbf{G}}^1 = (3, 1, 1)$ and $\bar{\mathbf{G}}^2 = (2, 2, 1)$. The partition $\bar{\mathbf{G}}^1$ provides two server vectors: $\bar{\mathbf{S}}^1 = (1, 1, 3)$ and $\bar{\mathbf{S}}^2 = (1, 3, 1)$. $\bar{\mathbf{S}}^1$ is eliminated, since $S_2^1 < S_2^{\text{LB}}$. For $\bar{\mathbf{S}}^2$, the workload allocation problem is solved and the resulting throughput is less than the required throughput. The partition $\bar{\mathbf{G}}^2$ provides only one server vector: $\bar{\mathbf{S}}^1 = (1, 2, 2)$. The solution to the workload allocation problem gives a throughput that is less than the throughput requirement. Thus, there is no feasible solution at $N = 5$, $K = 5$, and so $(N^{\text{f}}, K^{\text{f}}) = (N^{\text{I}}, K^{\text{I}}) = (5, 6)$.

(3b,c) Search over $N > N^{\text{f}}$.

We first consider $N = N^{\text{f}} + 1 = 6$. To find the smallest feasible K, we start with $K = 5$, giving $z(6, 5) = 28{,}600$. When $K = 5$, the partitions $\bar{\mathbf{G}}^1 = (3, 1, 1)$ and $\bar{\mathbf{G}}^2 = (2, 2, 1)$ are examined. A better feasible solution is found when the workload allocation problem is solved with $\bar{\mathbf{S}}^1 = (1, 2, 2)$, and so the incumbent solution is updated as $(N^{\text{P}}, K^{\text{P}}) = (6, 5)$.

We next decrease K by 1. This provides one partition $\bar{G}^1 = (2, 1, 1)$ and two resulting server vectors $\bar{S}^1 = (1, 1, 2)$ and $\bar{S}^2 = (1, 2, 1)$. No feasible solution is found for both $\bar{S}$'s: $\bar{S}^1$ is eliminated from further consideration since $S_2^1 < S_2^{LB}$, and $\bar{S}^2$ is eliminated since TH($\bar{S}^2$, $\bar{W}^*$) is less than the throughput requirement.

We now increase N by one unit at a time and, for each value of N, find the largest K such that $z(N, K) < z(N^p, K^p)$ and the solution is feasible. For $N = 7$, in order to have $z(N, K) < 28{,}600$, we must have $K \leq 4$, but this gives a throughput less than the requirement. Similarly, for $N = 8$, we must have $K \leq 4$, but this is infeasible too. A better feasible solution is found at $N = 9$, $K = 4$, with $\bar{S}^2 = (1, 2, 1)$. The incumbent solution is updated as $(N^p, K^p) = (9, 4)$.

We now try to decrease K. This results in $K < K^{LB}$.

(3d,e) Search over $K \geq K^f$. For $K = 6$, the largest N such that $z(N, 6) < z(N^p, K^p) = 25{,}400$ is less than N^{LB}.

(3f) The current incumbent solution, namely, $(N^p, K^p) = (9, 4)$, with $\bar{S}^* = (1, 2, 1)$ and $\bar{W}^* = (7.5, 15, 7.5)$, is optimal.

The search process for the example is illustrated graphically in figure 1.

4. Experimental results

We use five sets of parameters to illustrate the optimal algorithm for a single machine type. We use the following linear cost function for $z(N, K)$:

$$z(N, K) = (C_h + C_p + C_a) \cdot N + C_k \cdot K$$

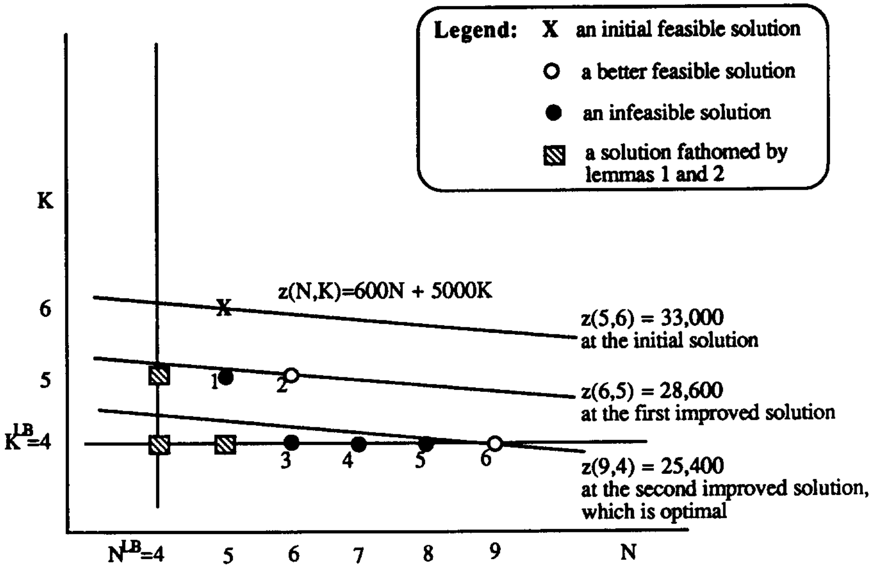

Figure 1. An example for the implicit enumeration procedure. The numbers near the circles indicate the sequence in which the solutions are evaluated. The optimum solution is indicated by "6."

where C_h, C_p, C_a and C_k are the annualized costs of a unit of work-in-process (WIP) inventory, a pallet, a stop-and-go AGV, and a machine, respectively. When the MHS is a loop-conveyor instead of AGVs, C_a is assigned a value of zero. Note that only the ratios of these cost parameters are relevant since the cost function is linear. To investigate various scenarios, we use different ratios for the five sets of cost parameters. The workload bounds are chosen arbitrarily, but are consistent with the other problem data. The problem data are presented in table 1.

The algorithm was coded in FORTRAN and run on an IBM 3090-600, using the VS-opt3 compiler. The following statistics were collected at termination of the algorithm for each problem: the optimum solution and its cost, the number of throughput computations, the number of workload allocation problems solved, and the CPU time. The number of throughput computations was recorded, since this consumed most the CPU time. The statistics are summarized in table 2.

Table 1. Five data sets.

Problem	M	d*960	TW	W_0	C_h	C_p	C_a	C_k
A	3	100	30	8	100	500	0	5000
B	4	100	60	25	100	500	600	2000
C	5	150	80	48	1	500	0	1000
D	6	200	80	35	1	500	500	1500
E	8	100	80	18	100	500	500	2500

Problem	$\bar{L}$ (work load lower bound)	$\bar{U}$ (work load upper bound)
A	(5,10,5)	(10,15,20)
B	(5,10,15,15)	(10,40,30,40)
C	(5,10,15,15,5)	(30,40,30,40,50)
D	(5,5,5,5,5,5)	(40,40,40,40,40,40)
E	(5,10,15,15,1,10,5,1)	(10,40,30,40,50,20,10,40)

Table 2. Results of experiments with the optimal algorithm.

Problem	Optimum Solution (N, $\bar{S}$, $\bar{W}$)	Optimum Cost	Number of Throughput Computations	Number of Workload Allocation Problems	CPU Time (sec)
A	9, (1,2,1) (7.5,15,7.5)	25,400	24	8	.05
B	12, (1,2,2,4) (5,11.7,15,28.3)	32,400	74	18	.11
C	25, (2,3,3,6,3) (7.4,13.2,15,30.5,13.9)	29,525	940	448	4.20
D	28, (10,5,2,2,2,2) (40,17.84,5.54,5.54,5.54,5.54)	62,528	2285	343	17.28
E	17, (1,2,2,2,2,2,1,1) (5,12.3,15,15,12.3,12.3,5,3.1)	51,200	25	13	.11

The results show that CPU time is sensitive to the throughput and the total workload. This follows, since a larger aggregate workload ($d \cdot \mathrm{TW}$) necessitates more servers, which in turn increases the number of partitions to be evaluated. The longest CPU time (17.28 seconds) was observed for problem D, which had the largest d (200/960) and the largest TW (80).

5. Extensions

We now consider more general versions of Problem Pl in which we relax some of the assumptions made in section 2. We first relax the assumption that a pallet carries only one part. When the parts are small, a pallet can carry a batch of parts; thus, the batch size may be another decision variable. Under Q-part transfer (where Q is the batch size), the Q parts are processed consecutively at the same machine. Thus, the Q units can be viewed as one "part" of a new product type whose total workload is $Q \cdot \mathrm{TW}$. The workload and throughput parameters in Pl must be scaled accordingly. The cost associated with pallets in the objective function should reflect the WIP inventory cost for Q parts instead of one part per pallet. We assume that the expected material-handling time (W_0) remains the same regardless of the batch size. In other words, the speed of the handling equipment is unaffected by the weight of the pallets. Therefore, with Q-part transfer, the formulation of problem Pl is restated as follows:

Pl$^{\mathrm{Q}}$:

$$\text{Minimize} \quad z(N, K, Q)$$

subject to:

$$K = \sum_{i=1}^{M} S_i,$$

$$\mathrm{TH}(M, N, \bar{\mathbf{S}}, \bar{\mathbf{W}}) \geq d/Q, \tag{5}$$

$$\sum_{i=1}^{M} W_i = Q \cdot \mathrm{TW}, \tag{6}$$

$$Q \cdot L_i \leq W_i \leq Q \cdot U_i, \qquad i = 1, \ldots, M. \tag{7}$$

We use the examples in table 1 to study the effect of Q on the minimum cost. For Q = 1, 2, 3, 4, 5, 10, 15, 20, and 30, we solved problem Pl$^{\mathrm{Q}}$. The results are shown in figure 2. We assume that WIP inventory costs are linear with respect to Q. The optimum batch size is determined by trading off three cost terms: material-handling cost, machine cost, and WIP inventory cost. A large Q reduces the number of material moves but increases WIP inventory cost. To compensate for this, the optimal value of N, the number of pallets

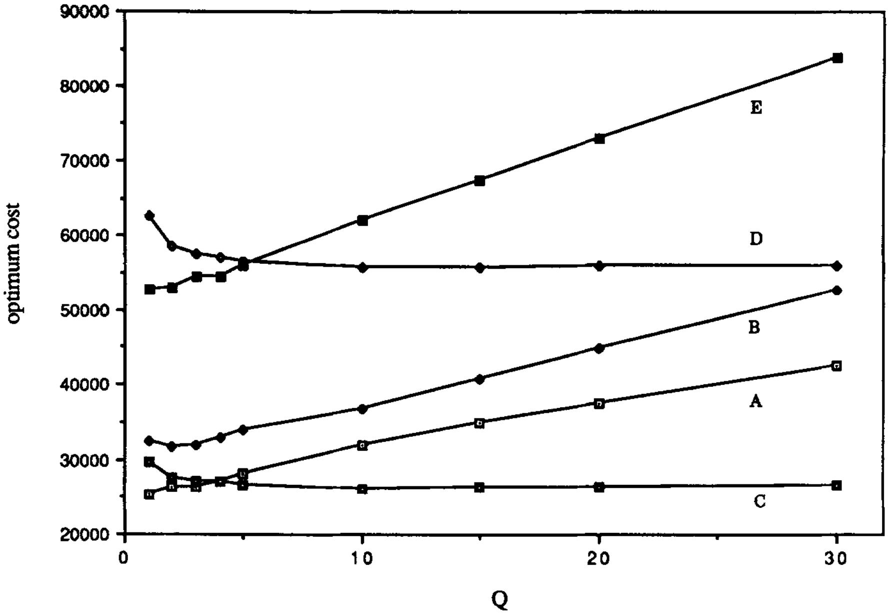

Figure 2. Effect of transfer batch size Q on optimum cost.

(and stop-and-go AGVs) in the system usually declines. As a result, servers may be idle for a long time while waiting for a pallet to arrive. This, in turn, increases the number of machines required to achieve the desired throughput. We observed that the total cost function is unimodal in Q. Based on this observation, a simple line search procedure for the optimal Q could be adequate to solve the problem.

We now consider the case of C machine types, and assume $Q = 1$ for ease of presentation. Let TW_c be the total workload for machine type c, i.e., the mean service time demanded by a part from machine type c. Also let:

K_c = number of machines of type c,
M_c = number of stations of type c,
S_{ci} = number of machines at station i of type c,
W_{ci} = workload for station i of type c,
L_{ci} = lower bound on workload for station i of type c,
U_{ci} = upper bound on workload for station i of type c.

The optimum configuration and workload allocation problem becomes

P1^C:

Minimize $\quad z(N, K_1, K_2, \ldots, K_C)$

subject to:

$$K_c = \sum_{i=1}^{M_c} S_{ci}, \qquad c = 1, \ldots, C, \tag{8}$$

$$\mathrm{TH}(M, N, \bar{\mathbf{S}}, \bar{\mathbf{W}}) \geq d, \tag{9}$$

$$\sum_{i=1}^{M_c} W_{ci} = \mathrm{TW}_c, \qquad c = 1, \ldots, C, \tag{10}$$

$$L_{ci} \leq W_{ci} \leq U_{ci}, \qquad i = 1, \ldots, M_c, \quad c = 1, \ldots, C, \tag{11}$$

where $z(N, K_1, K_2, \ldots, K_C)$ is any cost function that increases with N and K_c for any c; $M = \sum_{c=1}^{C} M_c$; $\bar{\mathbf{S}} = (\bar{\mathbf{S}}_c)$ with $\bar{\mathbf{S}}_c = (S_{c1}, \ldots, S_{cM_c})$; and $\bar{\mathbf{W}} = (\bar{\mathbf{W}}_c)$ with $\bar{\mathbf{W}}_c = (W_{c1}, \ldots, W_{cM_c})$.

The solution procedure for problem P1$^{\mathrm{C}}$ is as follows. We consider each machine type, c, in isolation and use the solution procedure developed in section 3, to obtain the optimal values of K_c^*, $\bar{\mathbf{S}}_c^*$, N_c^*, and $\bar{\mathbf{W}}_c^*$ with a throughput requirement of d. We next consider the overall system with the C machine types, with a server vector given by the $\bar{\mathbf{S}}_c^*$ values, and a workload allocation given by the $\bar{\mathbf{W}}_c^*$ values found above. This system is evaluated for each N until the throughput is greater than or equal to d. This gives us an initial feasible solution. Let C_{UB} denote the cost of this solution. Clearly a lower bound on the cost, C_{LB}, is given by K_c^{LB}, $c = 1, \ldots, C$, and N^{LB}, which are obtained in the same way as for the single-machine-type case.

We now generate all possible combinations of $(K_1, \ldots, K_C, N)$ that have cost between C_{UB} and C_{LB}, rank these combinations in decreasing order of cost, and implicitly enumerate the candidates in this list using a bisection search. For each candidate examined, we obtain the optimal configuration and the corresponding workload allocation by solving problem P2$^{\mathrm{C}}$, which is a generalization of problem P2:

P2$^{\mathrm{C}}$:

$$\text{Maximize} \qquad \mathrm{TH}(\bar{\mathbf{S}}, \bar{\mathbf{W}})$$

subject to constraints (8), (10), (11) of P1$^{\mathrm{C}}$.

Lemmas 1 through 4 and the Remark can be generalized and applied to problem P2$^{\mathrm{C}}$. It can easily be shown that the maximum throughput remains monotonic with respect to N and K_c for $c = 1, \ldots, C$ (cf. lemmas 1 and 2). Also lemmas 3 and 4 and the Remark hold for stations of the same machine type.

If the resulting throughput is feasible, then we can reduce the number of candidates that still need to be examined by half, and continue the bisection search. On the other hand, if the candidate being examined does not provide a feasible solution, then we cannot reduce the number of remaining candidates by half. However, we can still eliminate the current configuration and other configurations that are infeasible because of the monotonicity of the throughput function with respect to N and K_c, $c = 1, \ldots, C$. We then resume the bisection search on the remaining candidates. Since we use a bisection search, the workload allocation problem may need to be solved only for a relatively small number of candidates.

6. Conclusions

In this article, we considered the problem of finding the minimum cost configuration for an FMS subject to a constraint on throughput when there is some flexibility in allocating the workload among stations. The cost function includes the cost of machines, as well as the costs of material-handling equipment and work-in-process inventory.

We presented an implicit enumeration procedure for the problem with one machine type. We developed several fathoming methods to reduce the number of system configurations that must be evaluated. Computational experience with the algorithm suggests that problems of moderate size can be solved optimally within 20 seconds of CPU time on the IBM 3090-600 mainframe.

We also outlined an optimal algorithm for the more general problem with multiple machine types. Further research is needed to develop efficient heuristics for this problem.

Acknowledgment

We thank Dr. Yeong-Dae Kim for providing the FORTRAN code that generates partitions for a given number of machines. The helpful comments and suggestions of the editor, the associate editor, and the referees are greatly appreciated.

Appendix 1

In this appendix, we give a simple example to show how precedence constraints among operations influence the upper and lower bounds on workloads. Consider a flexible-flow system with one machine type that is capable of processing all 30 operations for a given product. However, because of tool magazine constraints or limits on the number of components that can be located nearby, only 20 operations can be performed by a given machine at any point in time. Suppose the precedence relations specify that operation j must be performed before operation k if $j < k$ (i.e., serial precedence structure). It is clear that two stations are sufficient. For simplicity, we will assume that two stations are used.

Assume that the processing time of operation i is i time units. The total workload per unit is 465 minutes. If we were to ignore the precedence constraints discussed above, the upper and lower bounds would be

$$L_1 = L_2 = \sum_{i=1}^{10} t_i = 55,$$

$$U_1 = U_2 = \sum_{i=11}^{30} t_i = 410.$$

On the other hand, if precedence constraints are considered, a little logic will show that

$$L_1 = \sum_{i=1}^{10} t_i = 55,$$

$$U_1 = \sum_{i=1}^{20} t_i = 210,$$

$$L_2 = \sum_{i=21}^{30} t_i = 255,$$

$$U_2 = \sum_{i=11}^{30} t_i = 410,$$

which are quite different from the bounds given above.

It is important to note that a continuous workload allocation satisfying the latter set of contraints may not be achievable, given the actual operation times. However, such an allocation is much more likely to be achievable (with respect to precedence constraints) than that obtained using the looser bounds.

Appendix 2

In this appendix, we provide proofs of lemmas 1 through 4. Lemmas 1 and 2 permit us to eliminate some dominated (N, K) pairs, while lemmas 3 and 4 permit us to reduce the number of $\bar{\mathbf{S}}$'s that must be considered for each undominated (N, K) pair.

Lemma 1. If $\mathrm{TH}(\bar{\mathbf{S}}^*(N, K), \bar{\mathbf{W}}^*(N, K)) < d$, then $\mathrm{TH}(\bar{\mathbf{S}}^*(N, K-1), \bar{\mathbf{W}}^*(N, K-1)) < d$.

Proof. We will prove this by contradiction. Suppose $\mathrm{Th}(\bar{\mathbf{S}}^*(N, K), \bar{\mathbf{W}}^*(N, K)) < d$ and $\mathrm{TH}(\bar{\mathbf{S}}^*(N, K-1), \bar{\mathbf{W}}^*(N, K-1)) \geq d$. Add to $\bar{\mathbf{S}}^*(N, K-1)$ one server in the ith station to give a total of K servers. Then, it follows from the results on the monotonicity of throughput with increasing service rates (Suri, 1984) that $\mathrm{TH}(\bar{\mathbf{S}}^*(N, K-1) + \mathbf{e}_i, \bar{\mathbf{W}}^*(N, K-1)) \geq \mathrm{TH}(\bar{\mathbf{S}}^*(N, K-1), \bar{\mathbf{W}}^*(N, K-1)) \geq d$ where $\mathbf{e}_i$ is a unit server vector with all elements zero except the ith element, which is set to 1. By definition of $\bar{\mathbf{S}}^*(N, K)$ and $\bar{\mathbf{W}}^*(N, K)$, $\mathrm{TH}(\bar{\mathbf{S}}^*(N, K), \bar{\mathbf{W}}^*(N, K)) \geq \mathrm{TH}(\bar{\mathbf{S}}^*(N, K-1) + \mathbf{e}_i, \bar{\mathbf{W}}^*(N, K-1)) \geq d$. This contradicts our original assumption. ■

Lemma 2. If $\mathrm{TH}(\bar{\mathbf{S}}^*(N, K), \bar{\mathbf{W}}^*(N, K)) < d$, then $\mathrm{TH}(\bar{\mathbf{S}}^*(N - 1, K), \bar{\mathbf{W}}^*(N - 1, K)) < d$.

Proof. The proof is similar to that of the previous lemma. ■

Lemma 3. If $U_i \leq L_k$ for any i and k, then we only need to consider $\bar{\mathbf{S}}$ such that $S_i \leq S_k$.

Proof. The product-form CQN under consideration consists of one delay node (the MHS station) and M multiple-server stations. The delay node also can be viewed equivalently as a multiple-server node with N servers, so there are no queueing delays. Thus, the CQN can be treated as a network where all stations have one or more servers. Shanthikumar and Yao (1988) show that the throughput function $\mathrm{TH}(\bar{\mathbf{S}}, \bar{\mathbf{W}})$ of the multiple-server product-form CQN is decreasing in transportation. That is, interchanging S_i and S_k so that $S_i \leq S_k$ whenever $W_i \leq W_k$ may increase, and does not decrease, the throughput. This rearrangement changes neither K nor $z(N, K)$. Thus, if such a rearrangement is possible, it is preferable to do so. When $U_i \leq L_k$, $W_i \leq W_k$ for any feasible $\bar{\mathbf{W}}$. Therefore, any $\bar{\mathbf{S}}$'s not satisfying the relationship in the lemma are dominated. ■

Lemma 4. If $L_i \leq L_k \leq U_i \leq U_k$, then we only need to consider $\bar{\mathbf{S}}$ such that $S_i \leq S_k$.

Proof. For any feasible $\bar{\mathbf{W}}$, we have two cases.

Case 1. $W_i \leq W_k$. From lemma 3, we only need to consider $\bar{\mathbf{S}}$ such that $S_i \leq S_k$.

Case 2. $W_i > W_k$. Consider $\bar{\mathbf{W}}' = (W_1', \ldots, W_{M'})$, which is obtained by interchanging W_i and W_k of $\bar{\mathbf{W}}$ while keeping the other workloads fixed. This $\bar{\mathbf{W}}'$ is feasible, since $L_i \leq L_k \leq W_k < W_i \leq U_i \leq U_k$ implies that $L_i \leq W_k \leq U_i$ and $L_k \leq W_i \leq U_k$, or equivalently, $L_i \leq W_i' \leq U_i$ and $L_k \leq W_k' \leq U_k$ by the definition of $\bar{\mathbf{W}}'$. Hence, by applying the result of case 1 to $\bar{\mathbf{W}}'$, we prove this lemma. This argument is valid, since the throughput function is permutation invariant, i.e., $\mathrm{TH}(\bar{\mathbf{S}}, \bar{\mathbf{W}}) = \mathrm{TH}(\pi(\bar{\mathbf{S}}), \pi(\bar{\mathbf{W}}))$ for any permutation π. ■

Appendix 3.

The following algorithm can be used to find a good initial feasible solution to the constrained workload allocation problem. We assume that the indices of the stations are arranged so that $S_1 \geq S_2 \geq \ldots \geq S_M$. We also assume that there is a feasible workload allocation (i.e., $\mathrm{TW} \leq \Sigma_i U_i$).

1. Find a balanced workload allocation, $\bar{\mathbf{W}}$. If it is feasible, then terminate. Otherwise, go to step 2.
2. Let $A = \{i | W_i > U_i\}$, $B = \{i | W_i < L_i\}$, $S_A = \sum_{i \in A} (W_i - U_i)$, $S_B = \sum_{i \in B} (L_i - W_i)$.
   Reset W_i to U_i for all $i \in A$ and to L_i for all $i \in B$. If $S_A - S_B > 0$ (less than the total workload is allocated), go to step 3. If $S_A - S_B < 0$ (more than the total workload TW is allocated), go to step 4. Otherwise, terminate.

3. Reallocate $S_A - S_B$ by assigning as much additional workload as possible to stations $1, \ldots, M$ in sequence while maintaining feasibility. Terminate whenever a feasible reallocation has been found.
4. Reduce the workloads at stations $M, \ldots, 1$ in sequence while maintaining feasibility, until a total reduction of $S_B - S_A$ has been achieved.

The rationale for steps 3 and 4 is a result of Shanthikumar and Yao (1988) that for the multiple-server product-form CQN, throughput is increased by assigning more workload to a station with a larger number of servers.

References

Avriel, M., *Nonlinear Programming Analysis and Methods*, Prentice-Hall, Inc., Englewood Cliffs, NJ (1976).

Dallery, Y. and Frein, Y., "An Efficient Method to Determine the Optimal Configuration of a Flexible Manufacturing System," *Proceedings of the 2nd ORSA/TIMS Conference on Flexible Manufacturing Systems*, K.E. Stecke and R. Suri (eds.) Elsevier Science Publishers, B.V., Amsterdam, pp. 269–282 (1986).

Gordon, W.J. and Newell, G.F., "Closed Queueing Networks with Exponential Servers," *Operations Research*, Vol. 15, pp. 252-267 (1967).

Kleinrock, L. *Queueing Systems I and II*, John Wiley and Sons, Inc., New York, NY (1976).

Lee, H.F., Srinivasan, M.M., and Yano, C.A., "Some Characteristics of Optimal Workload Allocation for Closed Queueing Networks," *Performance Evaluation*, Vol. 13, No. 1 (1991).

Muntz, R.R. and Wong, J.W., "Asymptotic Properties of Closed Queueing Network Models," *Proceedings of the 8th Annual Princeton Conference on Information Sciences and Systems*, Princeton University, Princeton, NJ (1974).

Saigal, R., "On the Convergence Rate of Algorithms for Solving Equations That Are Based on Methods of Complementary Pivoting," Vol. 2, pp. 108-124 (1977).

Shanthikumar, J.G. and Yao, D.D., "Optimal Server Allocation in a System of Multi-Server Stations," *Management Science*, Vol. 33, No. 9, pp. 1173-1180 (1987).

Shanthikumar, J.G. and Yao, D.D., "On Server Allocation in Multiple Center Manufacturing Systems," *Operations Research*, Vol. 36, No. 2, pp. 333–342 (1988).

Solberg, J.J., "A Mathematical Model of Computerized Manufacturing Systems," *Proceedings of the 4th International Conference of Production Research*, Tokyo, Japan (August 1977).

Stecke, K.E., "A Hierarchical Approach to Solving Machine Grouping and Loading Problems of Flexible Manufacturing Systems," *European Journal of Operational Research*, Vol. 24, pp. 369-378 (1986).

Stecke, K.E. and Solberg, J.J. "The Optimality of Unbalancing Both Workloads and Machine Group Sizes in Closed Queueing Networks of Multi-Server Queues," *Operations Research*, Vol. 33, No. 4, pp. 882-910 (1985).

Suri, R., "A Concept of Monotonicity and Its Characterization for Closed Queueing Networks," *Operations Research*, Vol. 33, No. 3, pp. 606-624 (1985).

Suri, R. and Hildebrant, R.R., "Modeling Flexible Manufacturing Systems Using Mean-Value Analysis," *Journal of Manufacturing Systems*, Vol. 3, No. 1, pp. 27–38 (1984).

Vinod, B. and Solberg, J.J., "The Optimal Design of Flexible Manufacturing Systems," *International Journal of Product Research*, Vol. 23, No. 6, pp. 1141-1151 (1985).

Yao, D.D., "Some Properties of the Throughput Function of Closed Networks of Queues," *Operations Research Letters*, Vol. 3, No. 6, pp. 313-317 (1985).

Yao, D.D. and Shanthikumar, J.G. "Some Resource Allocation Problems in Multi-Cell Systems," *Proceedings of the 2nd ORSA/TIMS Conference on Flexible Manufacturing Systems*, K.E. Stecke and R. Suri (eds.) Elsevier Science Publishers, B.V., Amsterdam, pp. 245–256 (1986).