Journal of the Korean
OR/MS Society
Vol. 13 No. 1 June 1988

－Research Paper－

An Allocation Problem in a Certain Class of Flexible Manufacturing Systems

Kim, Sung Chul*

Abstract

We study the optimal allocation of machines and pallets in a class of manufacturing systems. The FMS is modeled as a closed queueing network with balanced loading of the stations. An Algorithm is developed, which exploits the properties of the throughput function and solves the allocation problem for increasing concave profit and convex cost. We also study the more general case of allocating machines and pallets among a set of FMSs. A dynamic programming approach is developed, which solves the problem with $0(M^3N^2)$ operations.

1. Introduction

Consider a manufaturing system which produces a variety of part types and volumes. All part types are classified and grouped as appropriate families of part types and form a set of individual cells(usually termed as "Group Technology" and "Cellular Manufacturing"). Each cell consists of an automated manufacturing system termed as a Flexible Manufacturing System(FMS).

A FMS consists of a set of work stations, and at each station, a set of parallel machines operate, which are quite versatile in functions. Another component of FMS is a pallet which is a device to carry the job through it's circulation within the sytstem.

Suppose the number of machines and the number of pallets(alternatively, the number of jobs to be circulated in the system) are decision variables. There is a profit obtained from the throughput, and a cost to install the machines and equip the system with pallets. Then, the objective is to maximize the net profit, i.e.,the throughput profit minus the cost of machines and pallets. In this case, an important problem which has a fundamental effect on the system performance is how to allocate the given resources among a set of cells to manufacture part families.

In section 2, we present some preliminaries of the problem which will be the building block for further analysis. In section 3, we study the simultaneous optimal allocation of machines and

*Duk Sung Women's University

pallets for a single cell. The first and second order properties of the throughput function are identified and exploited to develope an algorithm, which solves the allocation problem for increasing cancave profit function and convex cost functions. In section 4, we study the more general case of allocating machines and pallets among a set of cells. The optimal allocation problem is formulated as a dyamic programming problem, which can be solved $O(M^3N^2)$ operations, when M and N are the total number of machines and pallets respectively to be allocated. A numerical example is illustrated in section 5.

2. Preliminaries.

Consider a FMS modeled as a closed queueing network(Gorden & Newell 1967). The FMS has S work stations and N jobs. Station i has single server with service rate μ_i, $i=1, ..., S$. Job routing follows a Markov chain which is finite and regular(Kemeny & Snell 1976) with routing matrix γ_{ij}, where γ_{ij} is the routing probability from station i to station j. Let $\nu_i>0$, $i=1, ..., S$, be the solutions to the following equations :

$$\sum_{j=1}^{S} \nu_j \gamma_{ji} = \nu_i, \quad \sum_{i=1}^{S} \nu_i = 1, \quad i=1, \cdots, S. \tag{2.1}$$

Let $\rho_i=\nu_i/\mu_i$, $i=1, ..., S$, then ρ_i indicate the service intensity at station i, and we shall refer to it as the loading at station i, which can be interpreted as the average work requirement assigned to station i. Also, let X_i be the total number of jobs at station i, then the equilibrium joint queue-length distribution has the following product form :

$$P(X_1=n_1, \cdots, X_M=n_M) = G^{-1}(N) \Pi_{i=1}^{M} \rho_i^{n_i} \tag{2.2}$$

and G(N) is the normalizing constant : (2.3)

$$G(N) = \sum_{\sum n_i = N} \prod_{i=1}^{M} \rho_i^{n_i}$$

Define TH as the throughput function of the network, then TH can be derived as :

$$TH=G(N-1)/G(N) . \tag{2.4}$$

Let $L=\sum_{i=1}^{M} \rho_i$, the overall loading of a system. Suppose L is given and fixed, then the loading problem within the system concerns how to distribute given overall loading, i.e., all the operations of the product mix of part types, among stations to determine which machine will be able to perform each operation of part types within the associated technical constraints.

The optimality of balanced loading, $\rho=L/S$, $i=1, ..., M$, in system with single machine stations is a well known result(Shanthikumar 1982, Shanthikumar & Stecke 1986, Yao 1985), and one can easily derive the throughput function(TH) of the FMS under the balanced loading as :

$$TH=(SN/L)/(S+N-1). \tag{2.5}$$

3. A Single Cell.

Here we consider a single flexible manufacturing cell with the features described in section 2. The overall loading L is given. The number of machines, m, and the number of pallets, n, to be allocated are decision variables. To avoid degeneracy, we assume that there must be one machine and one pallet allocated to each cell. That is, the feasible ranges for m and n are, respectively, $1 \leqq m \leq M$ and $1 \leq n \leq N$, where M and N denote the total number of machines and pallets to be allocated. From (2.5), the throughput of the cell as a function of m and n can be expressed as :

$$TH(m,n)=(mn/L)/(m+n-1) \tag{3.1}$$

Let a(TH) be the profit function, and b(m), c(n) the cost functions. We assume that a(TH) is an increasing concave function of the throughput, and b(m) and c(n) are convex functions.

The first and second order properties of TH(m,n) are summarized in the following lemma.
Lemma 1. The throughput function, TH(m,n), of (3.1) is (i) increasing concave in m and n individually ; (ii) not jointly concave in (m,n) ; (iii) supermodular in (m,n).
Proof. Treat m and n as continuous variable, then the first and the second derivatives can be derived as follows :

$$TH'_m=(n/L)\ (n-1)/(n+m-1)^2,\ TH'_n=(m/L)(m-1)/(m+n-1)^2. \tag{3.2}$$
$$TH''_{mm}=-2(n/L)(n-1)/(m+n-1)^3,\ TH''_{nn}=-2(m/L)(m-1)/(m+n-1)^3. \tag{3.3}$$
$$TH''_{mn}=(mn+(m-1)(n-1))/(L(m+n-1)^3). \tag{3.4}$$

Then, (i) and (iii) follows from (3.2) and (3.4), respectively(for (iii), see Ross 1983) ; and (ii) is a result of $TH''_{nn}\cdot TH''_{mm}\leqq(TH''_{mn})^2$, which, from (3.3) and (3.4), is equivalent to

$$4mn(m-1)(n-1)\leq(mn+(m-1)(n-1))^2 \tag{3.5}$$

which is self-evident(We note that the same results can be proved by treating m and n as integers at the price of more algebra).

The optimal allocation problem can be formulated as follows :

$$\text{Max. } f(m,n)=a(TH(m,n))-b(m)-c(n), 1\leq m<M \text{ and } 1\leq n\leq N. \tag{3.6}$$

and for a given m, $1\leq m\leq M$, also define

$$B(m,n_m)=\text{Max.}_{1\leq n\leq N}\,(a(TH(m,n))-c(n)). \tag{3.7}$$

Since TH(m,n) is concave in n(Lemma 1) and a(TH) is increasing concave in TH, the composite function, a(TH(m,n)) is then concave in n. Therefore, the objective function in (3.7) is concave in n(recall that c(n) is convex in n as assumed, and hence -c(n) is concave), and $B(m,n_m)$ can be derived easily, given m. For instance, one can increase n, one unit at a time, and stop at the first time the objective function ceases to increase, or n=N, whichever comes first.

Once $B(m,n_m)$ is derived, the remaining part of the optimal allocation problem in (3.6) is to solve the following :

$$\text{Max. } f(m,n)=\text{Max.}_{1\leq n\leq M}\ (B(m,n_m)-b(m)). \quad (3.8)$$

Although $B(m,n)$ is concave in m(since $TH(m,n)$ is concave in m and hence $a(TH(m,n))$ is concave in m), $B(m,n_m)$ as defined in (3.7) need not be concave in m(recall from Lemma 1 that $TH(m,n)$ is not jointly concave in (m,n)). However, the problem in (3.8) can still be solved efficiently as follows :

Theorem 1. The optimal solution m^* to the problem in (3.8) can be derived as follows :

$$B(m^*,n_{m^*})-b(m^*)=\max._{1\leq m\leq m_0}(B(m,n_m)-b(m)), \quad (3.9)$$

where m_0 is the smallest integer that satisfies

$$B(m_0,\ n_{m_0})/m_0<b(m_0)-b(m_0-1). \quad (3.10)$$

To prove the above Theorem, we shall need the following Lemma and the proof is immediate and hence omitted.

Lemma 2. For a discrete function $\theta(k)$,

(i) $\theta(k)$ is convex(concave), if and only if $(n-m)\{(\theta(m)-\theta(1)\}\leq(\geq)\ (m-1)\{\theta(n)-\theta(m)\}$ for all $1\leq m\leq n$.

(ii) $\theta(k)$ is sublinear (superlinear), if and only if $\theta(n)-\theta(m)\ \leq(\geq)\ \{(n-m)/m\}\theta(m)$.

(iii) If $\theta(0)=0$ and convex(concave), then it also implies superlinear (sublinear).

Now we return to the original problem.

Proof. For $m\leq m_0$, $B(m,n_m)-b(m)\leq B(m^*,n_{m^*})-b(m^*)$ by definition. consider $m>m_0$, we have

$$B(m,n_m)-B(m_0,\ n_{m_0})\leq B(m,n_m)-B(m_0,n_m)\leq\{(m-m_0)/m\}B(m_0,n_m)$$
$$\leq\{(m-m_0)/m_0\}B(m_0,n_{m_0})\leq(m-m_0)\{b(m_0)-b(m_0-1)\}\leq b(m)-b(m_0),$$

where the lst and 3rd inequalities are due to $B(m_0,n_{m_0})=\max_n B(m_0,n)$, the 2nd inequality is due to Lemma 2, (ii) and (iii), the 4th inequality is due to (3.10) and the last inequality is due to lemma 2(i). Therefore, we have $B(m,n_m)-b(m)\leq B(m_0,n_{m_0})-b(m_0)\leq B(m^*,n_{m^*})-b(m^*)$. Hence (m^*,n_{m^*}) is optimal solution to (3.6). The line of proof is basically due to Shanthikumar and Yao (1987).

To summarize, the optimal allocation problem in(3.6) can be solved as follows : Incerease m one unit at a time. Every time solve (3.7) to get n_m and compute the value of the objective function in (3.8). If this value is greater than the value in memory, suppress the memory and replace it by this value(also keep in memory m and n_m) ; otherwise, discard this value. Repeat this procedure until either m_0 of (3.10) or M is reached. Whatever in memory is then the optimal solution.

Also, from the proof above, we have

$$B(m,n_m)-B(m_0,n_{m_0})<\{(m-m_0)m\}\times B(m_0,n_{m_0}), \quad (3.11)$$

for any m, m_0 ; $m\geq m_0$, i.e., $B(m,n_m)$, although not necessarily cancave in m, is indeed sublinear.

A special case of particular interest is the following profit function :

$$a(TH(m,n)) = g(L) \cdot TH(m,n), \tag{3.12}$$

where $g(L)$ is a function of L only(for instance, typically, $g(L)$ is an increasing function of L−the more work required by products, the higher their value-although we do not need this increasing property here). In this case, the search range of n in (3.7) can be reduced, and hence the computation for the jointly optimal solution of m and n.

Corollay 1. If the profit function takes the form of (3.12), the objective function in (3.7) is supermodular(see Lemma 1(iii)). The conclusion then follows, since n_m is 'isotone ' in m.

4. A System of Cells.

Now suppose we have a set of cells, $j=1, \ldots ,J$. Each cell has the features introduced in the last section. Let $\underline{m}=(m_j)_{j=1}^{J}$ and $\underline{n}=(n)_{j=1}^{J}$ denote the vectors of decision variables. Let $\pi_j(m_j,n_j)$ be the net profit function(refer to the last section ; here, however, we do not require the concavity of π_j for either m_j or n_j), for cell j, $j=1, \ldots ,J$. The objective is to maximize the total net profit : $\Gamma(m,n) = \sum_{j=1}^{J} \pi_j(m_j, n_j)$.

We use a dyamic programming formulation. Let

$$\Gamma_k(M_k,N_k) = \text{Max.} \sum_{j=1}^{k} \pi_j(m_j,n_j), \tag{4.1}$$

where the maximum is taken over all m_j's and n_j's such that $\sum_{j=1}^{k} m_j = M_k$ and $\sum_{j=1}^{k} n_j = N_k$, for given M_k and N_k. Then, from the principle of optimality, we have for $k=2, \ldots ,J$,

$$\Gamma_k(M_k,N_k) = \text{Max.}_{\substack{1\le m_k \le M_k-k+1 \\ 1\le n_k \le N_k-k+1}} \{\pi_k(m_k,n_k) + \Gamma_{k-1}(M_k-m_k, N_k-n_k)\}, \tag{4.2}$$

for $M_k=k, \ldots, M-J+k$ and $N_k=k, \ldots ,N-J+k$. The boundaries are :

$$\Gamma_1(M_1,N_1) = \pi_1(M_1,N_1),\ 1\le M_1 \le M-J+1 \text{ and } 1\le N_1 \le N-J+1. \tag{4.3}$$

And the optimal solution can be derived from $\Gamma_J(M,N)$. It is easy to verify that the number of operations of the dynamic programming recursion is $O(M^3N^2)$.

When there are costs(machine and pallet allocation costs) in the net profit function, the model in section 3 should be first applied to derive the optimal (m_j^*, n_j^*) for each individual cell.If $\sum_{j=1}^{J} m_j^* \le M$ and $\sum_{j=1}^{J} n_j^* \le N$, then these solutions remain jointly optimal. Otherwise, the dynamic programming model should be used to derive the joint optimal allocation. In this case, m_j^* and n_j^* serve as upper bounds for cell j, and should be included into the recursions of (4.2).

Note that the throughput function is symmetric in m and n ; but normally one can expect the cost of a machine is higher than the cost of a pallet, i.e., $b(m) \ge c(n)$ for $m=n$. These can be exploited to reduce the range of n_k in (4.2) to $n_k \ge m_k$, and $N_k \ge M_k$, and hence reduce the computation requirement.

5. Numerical Example.

Consider the optimal allocation of 9 machines and 19 pallets among 4 cells(i.e., $M=9$, $N=19$ and $J=4$), with loading $L_1=1$, $L_2=5$, $L_3=6$ and $L_4=14$. The profit and cost functions are as follows :

$$a_j(TH)=5000+(100/j)\cdot\log(mn^3)\cdot TH_j,$$
$$b_j(m)=2.5m\cdot(j^2L_j)+800/m^3,$$
$$c_j(n)=(\log L_j)^{1.5}n^2,$$

for $j=1,2,3$ and 4(We note these functions are selected for purpose of illustration only, they may not have any physical interpretations).

We first solve the optimal allocation separately for each individual cell. The results are presented in table 5.1. Note that the total number of machines and the total number of pallets required exceed $M=9$ and $N=19$ respectively.

We then solve the joint optimal allocation problem using the dynamic programming approach of the last section under the constraints $M=9$ and $N=19$. The results are summarized in Table 5.2.

Table 5.1. The optimal allocation for each cell.

cell	m_j^*	n_j^*	net profit(π_j^*)
1	6	16	9602.8
2	3	5	4896.3
3	2	3	4641.7
4	2	1	3777.0

Table 5.2. The joint optimal allocation for the system of 4 cells.

cell	m_j^*	n_j^*	net profit(π_j^*)
1	5	16	8951.9
2	2	1	4804.9
3	1	1	4062.6
4	1	1	3635.7
total	9	19	21455.1

6. Conclusion.

We studied the optimal allocation of machines and pallets in a class of manufacturing systems. The FMS is modeled as a closed queueing network with balanced loading of the stations. An algorithm is developed, which exploits the properties of the throuhtput function and solves the allocation problem for increasing concave profit and convex cost. We also studied the more general case of allocating machines and pallets among a set of FMSs. A dynamic programming approach is developed.

References.

1. Gorden, W.J. and G.F. Newell, "Closed Queueing Networks with Exponential Servers," Operations Research 15, 252−27(1967).
2. Kemeny, J. G. and J.L. Cnell, Finite Markov Chains, Spring Verlag, New York, 1976.
3. Shanthikumar, J.G., "On the Superiority of Balanced Loading in Flexible Manufacturing System," unpublished manuscript, 1982.
4. Shanthikumar, J.G. and K.E. Stecke, "Reducing Work-in-Process Inventory in Certain Class of Flexible Manufacturing Systems," European Journal of Operational Research 26-2, 266-271(1986).
5. Shanthikumar, J.G. and D.D. Yao, "Optimal Server Allocation in a System of Multi-Server Stations," Management Science 33−9, 1173−1180 (1987).
6. Ross,S., Introduction to Stochastic Dynamic Programming, Academic, New York, 1983.
7. Yao, D. D., 'Some Properties of the Throughput Function of the Closed Networks of Queues," Operations Research letters 3, 313-318(1985).