DIO가공시스템에서의 기계배치문제

임준묵* · 황 학**

Machine Layout Problem in Direct-Input-Output Manufacturing System

Joon-Mook Lim, Hark Hwang

〈Abstract〉

This paper deals with a Direct-Input-Output Manufacturing System (DIOMS) which has a number of machine centers placed along a built-in automated storage/retrieval system (AS/RS). During its operations, the storage/retrieval(S/R) machine picks up a pallet from the pickup/deposit port of a machine center and then moves it either to an empty rack opening of the AS/RS for temporary storage or to place it on the port of another machine center for subsequent operation. The machine layout problem in DIOMS is formulated as an integer mathematical programming whose objective is to minimize the total expected distance of the loaded S/R machine during a production period. Recognizing the limit of the exact solution procedure(the Branch and Bound method), two improvement-type heuristics are proposed. One is based on the simulated annealing method and the other the pairwise interchange method. The validity of the heuristics is examined with example problems.

Keywords: layout problem, Direct-Input-Output Manufacturing System, built-in AS/RS, simulated annealing, pairwise interchange.

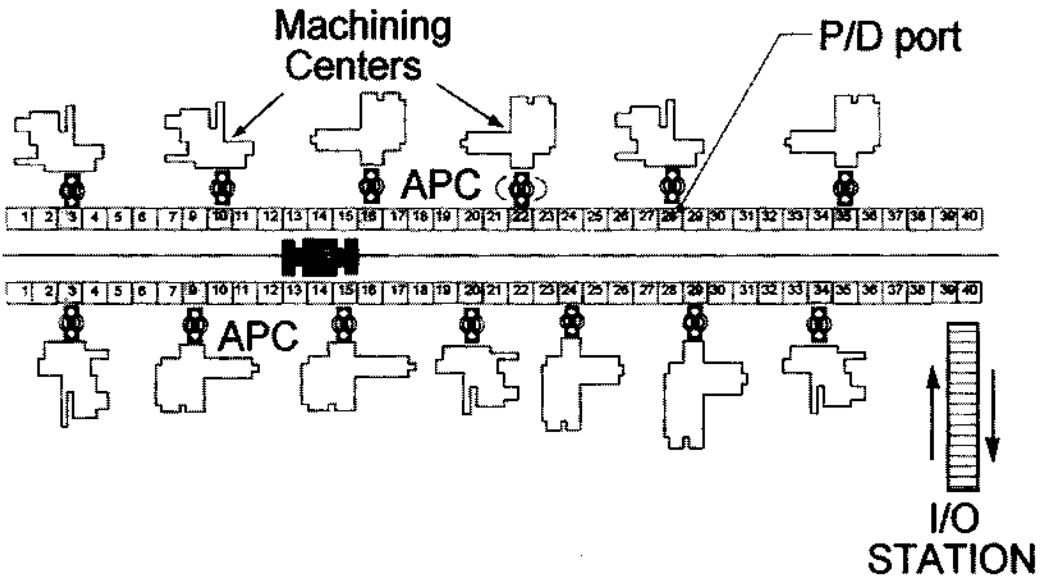
1. Introduction

Some modern manufacturing systems have workstation's directly integrated with centralized storage and handling system for work-in-process(WIP). The advantages of these integrated systems are better space utilization, real time inventory tracking, better production control, and flexibility in accommodating process changes. Kusiak(1985) illustrated a flexible manufacturing system(FMS) with built in automated storage/retrieval system(AS/RS) in which a stacker crane is designed to pick/place orders and load/unload machine tools. Chow(1986) analyzed an AS/RS which drives a "direct access handler" (DAH) in manufacturing assembly lines. DAH is similar to a robot device riding on a carriage that can move along a linear track and picks/delivers materials in a direct access fashion.

A domestic heavy manufacturing firm is currently operating a system which is similar to 〈Figure 1〉. The system

* Department of Industrial Engineering, Kangnung National University, Kangwondo, Kangnung, Jibyundong, 210-702, KOREA

** Department of Industrial Engineering, Korea Advanced Institute of Science and Technology, 373-1 Kusong-Dong Yusong-Gu Taejon 305-701, KOREA



⟨Figure 1⟩ Configuration of direct-input-output manufacturing system.

is called "direct-input-output manufacturing system" (DIOMS) and consists of 28 machine centers served by a storage/retrieval(S/R) machine with the storage capacity of 588 pallets. The machine centers are located adjacent to each other and placed evenly on both sides of the track. Ground level rack openings of the AS/RS become candidate sites for pickup and deposit(P/D) ports of machine centers. The operations are performed sequentially as the following manner: Parts placed on pallets enter the system through the input/output(I/O) station. The S/R machine picks up a pallet at the I/O station and places it on the pickup/deposit(P/D) port of the machine associated with its first operation. Pallets are fed into machine by the automatic pallet changer(APC). After the completion of machining operation, APC releases the pallet from the machine and put it back on the P/D port. When the S/R machine is to deliver a pallet for the subsequent operation and finds that the destination port is occupied with other pallet, the pallet will be placed in an empty rack opening of the central storage. It will be retrieved later when the destination port becomes empty. The pallet which has gone through all the required operations leaves the system via the I/O station.

From the above operation procedures, it can be concluded that the expected travel time of the S/R machine and thus the material handling cost depend on the relative locations of machine centers. In other words, the layout of machinecenters in DIOMS is closely related to its system efficiency. Here, the machine layout in DIOMS is to determine the relative locations of rack openings to be used as P/D ports of machine centers.

There exists a vast literature on the facility layout problem (Francis *et al.*(1992), Love *et al.*(1988)). Kusiak and Heragu(1987) published a review article on this problem. Heragu and Kusiak(1988) and Heragu(1992) dealt with the applicability of the quadratic assignment problem to flexible manufacturing system and studied four patterns of machine layout, i.e., circular single-row, linear single-row, linear double-row and multi-row. Afentakis(1989) and Kouvelis and Kim(1992) discussed the layout problem of the circular single row system in which the number of machines that part types cross in their manufacturing process is minimized. Houshyar and McGinnis(1990) proposed a heuristic procedure to assign the machines to locations along a straight track. Heragu and Kusiak(1990) and Kusiak(1990) suggested a knowledge-based system which combines the optimization with expert system approach. Jajodia *et al.*(1992) presented a simulated annealing method that considers the inter-cell and intra-cell layout problem in a cellular manufacturing environment. Recently, Kouvelis and Chiang(1992) dealt with the single row machine layout problem.

The layout problem in DIOMS is essentially a single row machine layout problem with the following characteristics: (i) the space required for each machine may not be identical. (ii) the number of candidate sites for machine centers are discrete and finite. (iii) the expected travel distance of pallet consists of two components, one between machine centers and the other between machine center and temporary storage area in the AS/RS. Due to the above characteristics, the results of the previous studies are not applicable to DIOMS. In this paper, we deal with the machine layout problem in DIOMS assuming that the physical configuration of the AS/RS and all the flow data of jobs(parts) are known.

The rest of this paper is organized as follows: In the next section, a mathematical model is formulated for the machine layout problem. An exact solution procedure and two heuristic algorithms are developed in section 3 and 4, respectively. A numerical example is solved to illustrate the developed algorithms in section 5. In section 6, computational experiences are given. Finally, conclusion is presented.

2. Mathematical Formulation

In order to model the machine layout problem of DIOMS, the following assumptions and notations are adopted:

Assumption:

(a) machines are to be located along both sides of the central storage. We consider only one side of the central storage for the convenience of analysis.

(b) machines are to be located in only one given direction.

(c) the shape and size of each machine are known.

(d) only the ground-level rack openings of AS/RS become the candidates for P/D ports with one opening for each machine.

(e) operation sequence and its processing time of each part type are known.

(f) part types and their production quantities are given.

(g) if DIOMS comes to the steady state, it is assumed that nearly all rack openings of AS/RS are occupied with work-in-processes. That is, we assume that randomized assignment policy is used for temporary storage in AS/RS.

(h) S/R machine is capable of handling one pallet at a time and can move simultaneously in both vertical and horizontal direction.

(i) each machine has a single load/unload buffer.

Notation:

i = index of machine type to be located, $i=1,\ldots,m$.

k = index of the ground-level rack openings of the AS/RS, $k=1,\ldots,n$, numbered to the right from the reference point.(refer to Figure 2)

r = size of rack opening(width).

f_{ij} = material flow between machine i and j(the number of pallets).

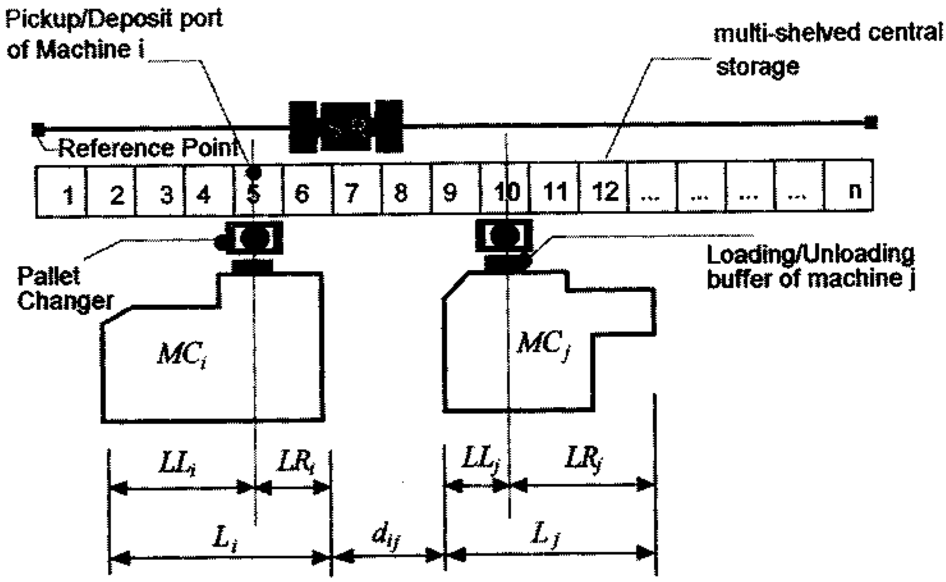
ET_k = expected travel distance between the ground-level rack opening k and temporary storage area.

p_i = probability that P/D port i is occupied.

d_{ij} = clearance requirement between machine i and j.

LL_i = horizontal length of machine i to the left from its loading/unloading buffer.

LR_i = horizontal length of machine i to the right from



〈Figure 2〉 Illustration of parameters for the machine layout problem.

its loading/unloading buffer.

L_i = length of machine $i(=LL_i + LR_i)$.

$$z_{ik} = \begin{cases} 1, & \text{if location } k \text{ is used as P/D port of machine } i, \\ 0, & \text{otherwise.} \end{cases}$$

$\lceil a \rceil$ = the smallest integer greater than or equal to a.

The system configuration and parameters are illustrated in 〈Figure 2〉. With the above assumptions and notations, we can formulate the machine layout problem(MLP) of DIOMS as follows:

Problem MLP:

$$\text{Min} \quad \sum_{i=1}^{m} \sum_{k=1}^{n} \sum_{j=1}^{m} \sum_{l=1}^{n} (1-p_j) f_{ij} r \, |k-l| \, z_{ik} z_{jl} +$$

$$\sum_{i=1}^{m} \sum_{j=1}^{m} \left| \sum_{k=1}^{n} p_j f_{ij} ET_k z_{ik} + \sum_{l=1}^{n} p_i f_{ji} ET_l z_{jl} \right| \tag{1}$$

$$\text{subject to} \quad \sum_{k=1}^{n} z_{ik} = 1 \quad \text{for all } i \tag{2}$$

$$\sum_{i=1}^{m} z_{ik} \leq 1 \quad \text{for all } k \tag{3}$$

$$\sum_{k=1}^{n-1} \sum_{l=k+1}^{n} r(l-k) z_{ik} z_{jl} \geq (LR_i + LL_j) + d_{ij} \text{ for all } i,j \tag{4}$$

$$\sum_{l=1}^{n-1} \sum_{k=l+1}^{n} r(k-1) z_{ik} z_{jl} \geq (LR_j + LL_i) + d_{ji} \text{ for all } i,j \tag{5}$$

$$LL_i \leq (k \cdot z_{ik} - 0.5) \cdot r \leq nr - LR_i \text{ for all } i,k \tag{6}$$

$$z_{ik} \in \{0,1\} \text{ for all } i,k \tag{7}$$

The objective of MLP is to minimize the total expected travel distance of the loaded S/R machine during the period of producing a prescribed number of parts. The first term of the objective function represents the sum of the expected distance that each part travels directly from one P/D port to another during its processing(direct flows). The remaining terms equal the expected total travel distance of parts which require temporary storage(indirect flows). Constraint (2) implies that each machine needs exactly one P/D port while (3) states that at most one machine can be assigned to a rack opening. Constraints (4) and (5) ensure the clearance requirement between two adjacent machines. Constraint (6) specifies the boundary condition for the layout problem.

Although decision variables are 0-1 integers and the objective function is quadratic form, MLP is a more complicated problem compared to the quadratic assignment problem(QAP).

Based on the randomized assignment policy for the temporary storage of pallet, appendix 1 shows a way to find approximate values of ET_k and p_i. In the next section, a procedure to find an exact solution will be presented utilizing the Branch and Bound technique.

3. Branch and bound solution procedure(exact solution procedure)

Let $\sigma_q = (\sigma(1), \sigma(2), ..., \sigma(q))$ be a sequence of machine type indices where $\sigma(i)$ is the index of machine type in the i-th position counting from the reference point. Also, let $L_q = (L(1), L(2), ..., L(q))$ be the location vector where $L(i)$ is the rack opening number assigned to the P/D port of machine $\sigma(i)$. Since separation of any two adjacent machines i and j by more than d_{ij} only increases the objective function value, MLP is equivalent to the m machine sequencing problem.

To utilize the branch and bound procedure, σ_q will be considered as a partial sequence. We shall use the word node as an alternative name for a partial sequence. By a completion of a partial sequence(or node) of size q we shall mean an assignment for which the first q integers are precisely those of the partial sequence. By the value of a partial sequence we shall mean a lower bound on the expected travel distance of the loaded S/R machine of all possible completion of the partial sequence. The value of a node will have the same meaning as the value of a partial sequence, and will be denoted by $\nu(\sigma_q)$. Let M_q be the set of machine indices associated with σ_q. Also, let $M_q^{'} = M - M_q$ where $M = \{1,2,3,...,m\}$. Obviously, number $(M_q^{'}) = m-q$. Suppose that the partial sequence $\sigma_q = (\sigma(1), ..., \sigma(q))$ is given, then $\nu(\sigma_q)$ can be represented by $\nu_1(\sigma_q) + \nu_2(M_q^{'}) + \nu_3(\sigma_q, M_q^{'})$, where $\nu_1(\sigma_q)$ = the minimum travel distance associated with the machines in σ_q, $\nu_2(M_q^{'})$ = a lower bound on the total travel distance associated with the machines in $M_q^{'}$, $\nu_3(\sigma_q, M_q^{'})$ = a lower bound on the total travel distance between all the pairs of machines, one σ_q and the other in $M_q^{'}$.

To find $\nu_1(\sigma_q)$, $\nu_2(M_q^{'})$ and $\nu_3(\sigma_q, M_q^{'})$, we introduce the following additional notations:

$ZA(M_q|\sigma_q) = ZAD(M_q|\sigma_q) + ZAI(M_q|\sigma_q)$, the minimum expected total distance of parts traveling between two machines in M_q under the condition of σ_q where $ZAD(M_q|\sigma_q)$ and $ZAI(M_q|\sigma_q)$ represent the direct and indirect flows, respectively.

$ZB(M_q^{'}) = ZBD(M_q^{'}) + ZBI(M_q^{'})$, a lower bound of the expected total distance of parts traveling between two machines in $M_q^{'}$ where $ZBD(M_q^{'})$ and $ZBI(M_q^{'})$ represent the direct and indirect flows, respectively.

$ZC((M_q|\sigma_q), M_q^{'}) = ZCD((M_q|\sigma_q), M_q^{'}) + ZCI((M_q|\sigma_q), M_q^{'})$, a lower bound of the expected total distance of parts traveling between all pairs of machines, one in M_q under the condition σ_q and the other in $M_q^{'}$ where $ZCD((M_q|\sigma_q), M_q^{'})$ and $ZCI((M_q|\sigma_q), M_q^{'})$ represent the direct and indirect flows, respectively.

(a) calculation of $\nu_1(\sigma_q)$

We need to find L_q which gives $ZA(M_q|\sigma_q)$. Let

$F(L_q) = ZAD(M_q|\sigma_q, L_q) + ZAI(M_q|\sigma_q, L_q)$

$$= \sum_{i,j \in M_q} \{(1 - p_{\sigma(j)}) \cdot f_{\sigma(i),\sigma(j)} \cdot r \cdot |L(i) - L(j)|\} + \sum_{i,j \in M_q} \{p_{\sigma(j)} \cdot f_{\sigma(i),\sigma(j)} \cdot (ET_{L(i)} + ET_{L(j)})\}$$

and

$\nu_1(\sigma_q) = \min_{L_q} \{F(Lq)\} = F(L_q^{*})$.

For the given σ_q, find L_q^{*} and $\nu_1(\sigma_q)$ as follows;

Step 0: $L(1) = \lceil LL_{\sigma(1)}/r + 0.499 \rceil$,

$L(i) = \lceil\{(L(i\text{-}1)\text{-}0.5)r + LR_{\sigma(i-1)} + d_{\sigma(i-1),\sigma(i)} + LL_{\sigma(i)}\}/r + 0.499\rceil$
for i=2,3,...,q.
$F = \infty$.

Step 1: Calculate $F(L_q)$ with σ_q and L_q.
Step 2: If $F(L_q) < F$ then F=$F(L_q)$.
Otherwise, set $L(1) \leftarrow L(1)+1$ and recalculate $L(i)$ for i=2,3,...,q.
Step 3: If $\{(L(q) - 0.5)r+LR_{\sigma(q)}\}/r<n$ then goto Step 1.
Otherwise let $\nu_1(\sigma_q) = F$ and STOP.

Initially, all the q machines are located as close as possible from the reference point, from which $F(L_q)$ is determined. And then we shift $L(1)$ to the right direction with an increment of one rack opening to the point until a further shift will only increase the value of $F(L_q)$.

(b) calculation of $\nu_2(M_q')$

i) A lower bound of $ZBD(M_q')$ may be obtained as follows. Let MF be the matrix of size $(m\text{-}q)\times(m\text{-}q)$ whose element of the i-th row and j-th column is $(1\text{-}p_j)f_{ij}$. Order the entries of the MF matrix in non increasing order except the diagonal elements to obtain a vector $VF = (VF_1, VF_2, ..., VF_s)$ where $s=(m\text{-}q)\times(m\text{-}q\text{-}1)$. Let $l_i, i\geq 1$ be the minimum length obtainable between the P/D port locations of any two machines in M_q', provided that $(i\text{-}1)$ number of machines in M_q' exist in between. We construct a vector VL of the size s such that the first $2(m\text{-}q\text{-}1)$ number of elements are l_2 which is followed by $2(m\text{-}q\text{-}2)$ number of l_3 and so until $(m\text{-}q\text{-}k)$= 1. Note that $l_2 < l_3 <,...,< l_{m-q}$. Then a lower bound is just the inner product of VF with VL.

ii) A lower bound of $ZBI(M_q')$ may be as follows. Suppose the machines in M_q are assigned from the reference point as close as possible. Let R_q be the length from the reference point to the right end side of $\sigma(q)$. Also, let $d1_{min} = \min_{i,j\in M_q'}\{d_{ij}\}$, $d2_{min} = \min_{i\in M_q, j\in M_q'}\{d_{ij}\}$, $LR_{min} = \min_{i\in M_q'}\{LR_i\}$, and $LL_{min} = \min_{i\in M_q'}\{LL_i\}$.

Step 1: Define $E_i = \sum_{j\in M_q'} (p_j f_{ij} + p_i f_{ji}),\ i \in M_q'$.

Step 2: Sort E_i's in non increasing order;
$E_{[1]} \geq E_{[2]} \geq,...,\geq E_{[m-q]}$.

Step 3: Let $L'_{[i]}$ = location of the i-th ranked machine;

$$L'_{[i]} = \lceil\{R_q + d2_{min} + LL_{min} + (i\text{-}1)(LR_{min} + d1_{min} + LL_{min})\}/r + 0.499\rceil$$
$$\text{for } (R_q + d2_{min} + LL_{min})/r \geq \frac{n}{2},\ i\in M_q'$$
$$= \left\lceil \frac{n}{2} \right\rceil,\ \text{for } (R_q + d2_{min} + LL_{min})/r < \frac{n}{2}$$

Step 4: $ZBI(M_q') = \sum_{i=1}^{m-q} E_{[i]} \cdot ET_{L'[i]}$.

(c) calculation of $\nu_3(\sigma_q, M_q')$

i) A lower bound of $ZCD((M_q \mid \sigma_q, L_q), M_q')$ may be obtained as follows. Let $R_p = \lceil(R_q + d2_{min} + LL_{min})/r + 0.499\rceil$ Note that R_p is the smallest number among the locations that a machine in M_q' is located next to R_q. And let $L'(j) = R_p$, for all $j\in M_q'$. Then,

$$ZCD((M_q \mid \sigma_q, L_q), M_q') =$$
$$\sum_{i\in M_q, j\in M_q'} [(1\text{-}p_j)\cdot f_{\sigma(i),j}\cdot r\cdot |L(i) - L'(j)|]$$
$$+ \sum_{i\in M_q', j\in M_q} [(1\text{-}p_{\sigma(j)})\cdot f_{i,\sigma(j)}\cdot r\cdot |L'(i) - L(j)|].$$

ii) A lower bound of $ZCI((M_q \mid \sigma_q, L_q), M_q')$ may be also obtained as follows.

Let $R_t = \left\lceil \frac{n}{2} \right\rceil$ if $R_q < \frac{n}{2}\cdot r - d2_{min} - LL_{min}$.

$= \lceil (R_q + d2_{min} + LL_{min})/r + 0.499 \rceil$. otherwise.

By assigning all the machines in M_q' on location R_t, i. e., $L'(j) = R_t$, for all $j \in M_q'$, $ZCI((M_q \mid \sigma_q, L_q), M_q')$ becomes

$$\sum_{i \in M_q, j \in M_q'} [p_j \cdot f_{\sigma(i),j} \cdot (ET_{L(i)} + ET_{L'(j)})] +$$

$$\sum_{i \in M_q', j \in M_q} [p_{\sigma(j)} \cdot f_{i,\sigma(j)} \cdot (ET_{L'(i)} + ET_{L(j)})].$$

While performing the branch and bound procedure, we can draw out the lower bound value at a node using the above results. This procedure is programmed using PASCAL language and executed on IBM PC(486) for the small sized problems. In section 5 and 6, performance results are presented with those of heuristic algorithms to be developed in next section.

4. Heuristic Algorithms

We propose two improvement-type heuristics, each of which consists of two phases: construction phase and improvement phase. Based on the initial assignment obtained in the construction phase, one adopts sidemove adjustments and the pairwise interchange and the other the simulated annealing method in the improvement phase.

4.1 Construction Phase

With p_i, $i=1,\dots,m$ given, let $DF_i(IF_i)$ be the sum of part flows directly(indirectly) coming to and outgoing from machine i during the scheduling period. Indirect flow are those flows which require central storage during its movement. DF_i and IF_i can be determined approximately as follows;

$$DF_i = \sum_{j=1}^{m} \{ (1-p_j)f_{ij} + (1-p_i)f_{ji} \} \quad (8)$$

and

$$IF_i = 2\sum_{j=1}^{m} (p_j f_{ij} + p_i f_{ji}). \quad (9)$$

Let $OF_i = DF_i + IF_i$. We rearrange OF_i in nonincreasing order such that $OF_{[1]} \geq,\dots,\geq OF_{[m]}$. The basic concept of the construction algorithm for an initial assignment is as follows: First, machine [1] is placed at the center position $CP = \lceil \frac{n}{2} \rceil$ of the racks. Then we determine the P/D port location of machine [2] examining points adjacent to machine [1] through evaluation of the objective function assuming that we have a two machine problem. Next, the location of machine [3] is determined in similar way and so on.

Construction algorithm for initial assignment

STEP 0 : Let $CP = \lceil \frac{n}{2} \rceil$, calculate $OF_i = DF_i + IF_i$ and set $SM' = \{1,\dots,m\}$, $SM = \phi$ and $k=1$.

STEP 1 : Arrange OF_i's in nonincreasing order i.e., $OF_{[1]} \geq OF_{[2]} \geq OF_{[3]} \geq,\dots,\geq OF_{[m]}$. Suppose s is the machine type index corresponding to [1].

STEP 2 : Place machine [1] to CP. Let $SM' = SM' - \{s\}$, $SM = SM \cup \{s\}$.

STEP 3 : Let $k = k + 1$. Let s be the machine index satisfying $OF_s = OF_{[k]}$. Determine whether enough space is available for the palcement of machine s to the left of the already assigned machines. If available, calculate the total cost $TC_l(s,SM)$ of the system consisting of $n(SM)$+1 number of machines. Otherwise, set $TC_l = \infty$. Similarly, we check the availability of space for machine s to the right direction of the already assigned machines and determine TC_r (s, SM), if available. Otherwise, set $TC_r = \infty$. When assigning the last machine, if available space does not exist in both directions, move the already assigned machines toward the right or left end, and check the availability of space for the final machine. And determine TC_l or TC_r, if available. Otherwise, set $TC_l = \infty$ and $TC_r = \infty$.

(i) If both TC_l and TC_r are infinite, this algorithm can not generate any feasible solution. Stop.

(ii) If one of them is finite, place machine s at the point adjacent to the machines $\in SM$. Go to step 4.

(iii) If both are finite, place the machine s at the position which incurs smaller total cost.

STEP 4 : Let $SM' = SM' - \{s\}, SM = SM \cup \{s\}$.

STEP 5 : If $k = m$, a feasible solution is obtained and stop. Otherwise, go to step 3.

4.2 Improvement algorithm I (sidemove adjustment and pairwise interchange)

With σ and L from the construction phase, the following two improvement schemes are adopted alternately until no improvement is made.

(a) adjustment by sidemove

With an initial assignment σ and L, we identify whether any rack opening exists which are not assigned to machines in σ. In case it exists, then we determine the objective function value resulted by one-step of L move toward available direction. If any cost reduction occurs, we make an adjustment in L for improvement.

(b) pairwise interchange

Note that not all pairs of machines can be candidate for pairwise interchange due to the clearance requirement.

STEP 1: Given σ and L coming from (a), identify all the candidate pairs of machines among the $m(m-1)/2$ number of pairs.

STEP 2: Find the pair of machines which results in the most reduction of objective function value. Update σ and L, and then go to step 1. If none, then stop.

4.3 Improvement algorithm II (Simulated annealing method)

Simulated Annealing(SA) is an algorithmic approach for the solution of optimization problems. The name of the algorithm derives from an analogy between solving optimization problems and simulating the annealing of solids (Eglese(1990)).

A description of the simulated annealing algorithm is given in ⟨Figure 3⟩.

Step 1: Get an **initial configuration** σ.
Step 2: Get an **initial temperature** $T > 0$.
Step 3:
- 3.1 For $1 \le i \le EL$ do
  - 3.1.1 Pick random neighbor σ' of σ.
  - 3.1.2 Let $\Delta = cost(\sigma') - cost(\sigma)$.
  - 3.1.3 If $\Delta \le 0$ then set $\sigma = \sigma'$.
  - 3.1.4 If $\Delta > 0$ then set $\sigma = \sigma'$ with probability $exp(-\Delta / T)$.
- 3.2 Set $T = cr \cdot T$.

Step 4: Return σ.

⟨Figure 3⟩ Simulated annealing algorithm.

Starting from an initial configuration(layout), the SA generates at random a new layout $\sigma' = \hat{X}$ in the neighborhood of the original configuration $\sigma = X$, and for which the change in the value of the objective function, $\Delta = Z(\hat{X}) - Z(X)$, can be quickly calculated. If the change represents a reduction in the value of the objective function, then the transition to the new layout is accepted. If the change represents an increase in the objective function value, then the transition to the new layout is accepted with a specified probability. The acceptance probability function usually takes the form $exp(-\Delta/T)$ where T is a control parameter. This provides a mechanism which enables an SA algorithm to avoid becoming trapped in a local minimum in its search for the global minimum.

The choices the designer of simulated annealing heuristic has to make can be classified into two classes, namely problem-specific and generic.

Problem specific parameters

(1) *configuration*: this is given by an assignment of the machines to the candidate locations i.e., $\sigma = Z = (z_{ij}$ $i=1,2,\ldots,m, j=1,2,\ldots,n)$.

(2) *neighborhood*: the neighborhood of a configuration consists of those configurations that result from the initial one by the interchange of the location of two machines.

(3) *cost*: the cost C of configuration $\sigma = X$ is the total expected value of (material flow × distance between machines) described in Eq.(1).

(4) *initial configuration*: it can be easily obtained by a random assignment or simple heuristic algorithm of machines to locations. In this paper, a construction heuristic solution is adopted as initial configuration.

Generic parameters

A choice of these parameters is referred to as a *cooling schedule*. In any implementation of a SA algorithm, a cooling schedule must be specified.

(1) *initial temperature(T_0)*: the initial temperature is chosen in such a way that a cost increasing transition would be accepted in the first stage of the annealing process with probability P_0. The value of T_0 is calculated according to the following formula;

$$P_0 \doteq exp(-\overline{\Delta}/T),$$

where $\overline{\Delta}$ is the mean cost increase calculated from performing a number of random transitions. The number of transitions attempted to calculate the $\overline{\Delta}$ is a fraction of the neighborhood size of the initial configuration.

(2) *epoch length(EL)*: the simulated annealing can be represented as a process which given a neighborhood structure attempts to transform the current configuration into one of its neighbors. This process can be modeled as a Markov chain. For each temperature value we get one such Markov chain. The length of each Markov chain, which is referred to as the epoch length, *EL*, is taken to be equal to a percentage of the total neighborhood.

(3) *cooling ratio*: the temperature should be decremented in such a way that we shall not have to employ very long Markov chains. This frequently used decrement rule specifies that: $T_k = cr \cdot T_{k-1}$, $k = 1,2,\ldots$ where cr is a constant(called by *cooling ratio*) less than 1.

(4) *frozen system(stopping criterion)*: the stopping criterion is a criterion which will specify when the system is frozen. The execution of SA algorithm is terminated if either of the following two conditions is satisfied. The first condition specifies that the execution is stopped if the optimal configuration found so far remains unchanged for a number of temperature reduction stages. The second condition specifies that the execution will halt if the number of attempted transitions drops below a certain point. For example, algorithm will stop if the number of

accepted transitions is less than a fraction of the total number of accepted transitions. In this paper, the second condition is adopted as stopping criterion.

5. Numerical Example

With the following example problem, we illustrate the heuristics developed.

The number of location sites(n) : 32
The height of central buffer(h) : 4
The number of machines(m) : 7
The size of rack opening(r) : 1.0
The length of scheduling period(SP) : 14000(unit time)
$d_{ij} = 1$ for all i,j.

The machine sizes to be located, parts, demands and processing times, and their processing sequences are given in Table 1, Table 2 and Table 3, respectively.

〈Table 1〉 Dimension of each machine.

Machine	1	2	3	4	5	6	7
L_i	1.0	4.5	4.0	5.0	2.0	2.5	3.0

(Note: $L_i = LL_i + LR_i(LL_i = LR_i = L_i/2)$)

〈Table 2〉 Parts to be manufactured, their demands and processing sequences.

Part	Demand(unit/scheduling period)	Machine processing sequence
A	100	1→2→5→6→4→3→1
B	300	1→3→4→6→2→7→1
C	500	1→4→5→6→1
D	400	1→2→3→7→5→1

〈Table 3〉 Processing time(unit time).

Part \ Machine	1	2	3	4	5	6	7
A	2	10	7	13	8	10	-
B	2	17	10	5	-	5	10
C	2	-	-	10	12	20	-
D	2	5	12	-	10	-	17

With the above data, approximate values of the parameters, $ET_k^{'}s$ and $pi's$, are obtained from the results in appendix 1 and listed in Table 4 and Table 5.

Also, f_{ij} are calculated from Table 2 and shown in Table 6.

〈Table 4〉 Estimated value of ET_k.

k	ET_k
1	15.578
2	14.664
3	13.796
4	12.984
5	12.234
6	11.546
7	10.921
8	10.359
9	9.859
10	9.421
11	9.046
12	8.734
13	8.484
14	8.296
15	8.171
16	8.109
17,...,32	$ET_{n-(k-1)}$

〈Table 5〉 Approxmate value of p_i.

machine i	p_i
1	0.2432
2	0.2836
3	0.2824
4	0.2391
5	0.5079
6	0.7639
7	0.4050

〈Table 6〉 Estimated value of f_{ij}.

from \ to	MC1	MC2	MC3	MC4	MC5	MC6	MC7
MC1	0	500	300	500	0	0	0
MC2	0	0	400	0	100	0	300
MC3	100	0	0	300	0	0	400
MC4	0	0	100	0	500	300	0
MC5	400	0	0	0	0	600	0
MC6	500	300	0	100	0	0	0
MC7	300	0	0	0	400	0	0

With the above data, the following results are obtained.

〈Table 7〉 Calculation of DF_i, IF_i and OF_i.

i	DF_i	IF_i	$OF_i(=DF_i+IF_i)$
1	1937.86	1324.29	3262.14
2	1087.88	1024.25	2112.12
3	1116.10	967.81	2083.90
4	1073.51	1452.97	2526.49
5	936.54	2126.93	3063.46
6	881.96	1836.07	2718.04
7	840.42	1119.17	1959.58

〈Table 8〉 Result of each iteration of the construction algorithm.

iterations	space availability		TC_l	TC_r	machine index (machine size)	location
	left	right				
1	-	-	-	-	1(1.0)	16
	○	○	2522.33	**2503.90**	5(2.0)	19
2	○	○	**14197.73**	15971.69	6(2.5)	13
3	○	○	32396.62	**31488.66**	4(5.0)	24
4	○	○	**41140.07**	46822.69	2(4.5)	8
5	○	○	61297.32	**60810.32**	3(4.0)	30
6	○	×	**87721.49**	infinite	7(3.0)	3

(Layout Results): σ=(1,5,6,4,2,3,7) L=(16,19,13,24,8,30,3) OBJ=87721.49

Table 7 and 8 list the result of each iteration of the construction procedure. Also Table 9 shows the final assignment by the application of two improvement algorithms.

〈Table 9〉 Performance of the solution algorithms.

Algorithms	iter.	Intrchngd pair	σ	L	OBJ	APD (%)
CA			(7,2,6,1,5,4,3)	(3,8,13,16,19,24,30)	87721.49	3.4
IA-I	0	initial	(7,2,6,1,5,4,3)	(3,8,13,16,19,24,30)	87721.49	3.4
	1	(3,4)	(7,2,6,1,5,**3**,**4**)	(3,8,13,16,19,**23**,**29**)	86029.67	1.4
	2	(2,7)	(**2**,**7**,6,1,5,3,4)	(**4**,**9**,13,16,19,23,29)	84836.66	0.0
IA-II	initial		(7,2,6,1,5,3,4)	(3,8,13,16,19,23,29)	87721.49	3.4
	results		(2,7,6,1,5,3,4)	(4,9,13,16,19,23,29)	84836.66	0.0
BB			(2,7,6,1,5,3,4)	(4,9,13,16,19,23,29)	84836.66	0.0

CA : Construction Algorithm　IA-I : Improvement Algorithm-I
IA-II : Improvement Algorithm-II　BB : Branch and Bound Procedure
APD : Average Percent Deviation from BB=$(Z_{HEU} - Z_{BB})/Z_{BB} \times 100(\%)$

〈Table 10〉 Data generation rules.

(1) machine size is 1.0 to 5.0($LL_i = LR_i = L_i/2$).
(2) number of part types(np) is 10 to 30.
(3) demand(D_i) for each part type is 50 to 300.
(4) processing time of a part type in a machine is distributed 2.0 to 30.0 (unit time).
(5) minimum distances(d_{ij}) lies in 1.0 to 2.0.
(6) number of location sites(n) is determined by $n = \left\lceil \sum_{i=1}^{m} L_i + 2 \times m \right\rceil$.
(7) height of rack storage(h) is 4 for all the problems.
(8) length of scheduling period(SP);

$$SP = \max_{i=1,\dots,m}(sp_i) \times (allowance\ factor)$$

wher $sp_i = \sum_{p=1}^{np} \sum_{l=1}^{n_{pi}} t_{pi}^{l} D_p$

np=the number of part types.
n_{pi}=the number of times which part type p is processed on machine i
t_{pi}^{l}=the l-th processing time for the part type p on machine i
D_p=demand for part type p.
allowance factor = 1.1.

6. Computational Experience

To investigate the validity of the developed heuristics, the performance of the heuristics are compared with those of the Branch and Bound procedure through randomly generated problems. The heuristic algorithm are programmed with PASCAL and tested on IBM-PC(486 DX2). Table 10 shows the way the test problems are generated. Table 11 gives the parameter values adopted for SA. These parameters are selected after many prior experimen-

〈Table 11〉 Values of generic parameters.

(1) Initial temperature(T_0) : set P_0=0.5 and calculate $T_0 = -\frac{\overline{\Delta}}{log_e(P_0)}\Big|_{P_0=0.5}$
(2) Cooling ration(r) : cr=0.9
(3) Epoch length(EL) : 20 % of all neighbourhoods i.e., $EL = 0.2 \times \frac{m(m-1)}{2}$
(4) Frozen system : algorithm will stop if a fraction of the accepted transition to the total attempted transitions is less than 0.05.

〈Table 12〉 Performance of the algorithms for small-sized problems.
(average percent deviation from BB and computation time)

	m	4	5	6	7	8	9
Proposed Algorithm	CA	1.72 (0.02)	3.49 (0.05)	7.20 (0.10)	6.60 (0.25)	8.65 (0.68)	8.04 (0.92)
	IA-I	**0.01** (0.07)	0.11 (0.11)	0.16 (0.30)	**0.35** (0.68)	**0.73** (1.50)	1.22 (2.40)
	IA-II	**0.01** (0.07)	**0.09** (5.55)	**0.15** (12.3)	0.39 (21.9)	0.83 (36.4)	**0.85** (46.5)
	BB	0.00 (0.13)	0.00 (0.33)	0.00 (3.65)	0.00 (40.4)	0.00 (466.)	0.00 (6752.)

Note : ( ) = average computational time(sec)

〈Table 13〉 Performance of the improvement heuristics for large-sized problems.
(average percent deviation from IA-I and computation time)

	m	10	15	20	30
Proposed Algorithm.	CA	8.19 (1.20)	9.20 (3.70)	10.57 (5.95)	11.44 (8.92)
	IA-I	0.00 (6.71)	**0.00** (90.6)	0.00 (540.)	0.00 (1204)
	IA-II	**-0.71** (87.7)	0.36 (721.)	**-0.86** (3103)	**-1.50** (5243)

Note : ( ) = average computational time(sec) tations. For each machine size $m(m=4,5,...,9)$ a set of 100 problems are generated following the rules of Table 10.

It can be observed that CA gives relatively good solution for small sized problems. For the problem size of m=9, CA gives the solution which is within about 8% from optimal solution with not much computation time. Two improvement heuristic seems almost equal in the performances for the problems we tested. For our interest, additional problems of size 10, 15, 20 and 30 are solved by heuristics. It turned out that the SA based heuristic outperforms the other as the size of problem increases(see Table 13).

7. Conclusion

In this paper, we deals with the machine layout problem in the Direct-Input-Output manufacturing system, a flexible manufacturing system with built in automated storage and retrieval system. A 0-1 integer mathematical program is developed with the objective of minimizing the total travel distance of parts during their processes. To find an optimal solution, the exact solution procedure is proposed based on the Branch and Bound method. Recognizing the limit of BB with large sized problems, we develop two improvement-type heuristics. Through the randomly generated test problems it is observed that the heuristics perform reasonably well compared to BB for the problems of up to size 9. Also, the SA based improvement algorithm is found to perform better than the other algorithm as the problem size increases.

【References】

[1] Afentakis, P., "A loop layout design problem for flexible manufacturing systems," International Journal of Flexible Manufacturing Systems, 1(2), 143-175, 1989.

[2] Chow, W. M., "An analysis of automated storage and retrieval systems in manufacturing assembly lines," IIE Transactions, June, 204-214, 1986.

[3] Eglese, R. W., "Simulated Annealing: A tool for Operational Research," European Journal of Operational Research, 46, 271-281, 1990.

[4] Francis, R. L., McGinnis, L. F. and White, J. A., Facility Layout and Location: An Analytical Approach.Prentice Hall, Englewood Cliffs, N.J., 1992.

[5] Gross, D. and Harris, C. M., Fundamentals of Queueing Theory. John Wiley & Sons, New York, 1985.

[6] Heragu, S. S. and Kusiak, A., "Machine layout problem in flexible manufacturing systems," Operations Research, 36(2), 258-268, 1988.

[7] Heragu, S. S., "Recent models and techniques for solving the layout problem," European Journal of Operational Research, 57, 136-144, 1992.

[8] Heragu, S. S. and Kusiak, A., "Machine Layout : an optimization and knowledge-based approach," International Journal of Production Research, 28(4), 615-635, 1990.

[9] Houshyar, A. and McGinnis, L. F., "A heuristic for assigning facilities to locations to minimize WIP travel distance in linear facility," International Journal of Production Research, 28(8), 1485-1498, 1990.

[10] Jajodia, S., Minis, I., Harhalakis, G. and Proth, J. M., "CLASS : Computerized LAyout Solutions using Simulated annealing," International Journal of Production Research, 30(1), 95-108, 1992.

[11] Kouvelis, P. and Chiang, W. C., "A simulated annealing procedure for single row layout problems in flexible manufacturing systems," International Journal of Production Research, 30(4), 717-732, 1992.

[12] Kouvelis, P. and Kim, M. W., "Unidirectional loop network layout problem in automated manufacturing system," Operations Research, 40(3), 533-550,

1992.
[13] Kusiak, A. and Heragu, S. S., "The facility layout problem," European Journal of Operational Research, 29, 229-251, 1987.
[14] Kusiak, A., "Material handling in flexible manufacturing system," Material Flow, 2, 79-95, 1985.
[15] Kusiak, A., Intelligent Manufacturing Systems. Prentice Hall International Editions, 1990.
[16] Love, R. F., Morris, J. G. and Wesolowsky, G. O., Facilities Location: Models & Methods, Elservier Science Publishing Co., Inc., 1988.

Appendix 1. Determination of the parameters

A1.1 Determination of ET_k

ET_k can be approximated by the expected one way travel distance of S/R machine when I/O point is located at rack opening k in the discrete rack opening system.

$$ET_k = \frac{1}{nh}\sum_{i=1}^{nh} y_i$$

where y_i is the one-way travel distance from rack opening k to rack opening i.

A1.2 Determination of p_i's

We assume that the capacity of central buffer(AS/RS) is large enough to store work-in processes(WIP) generated during the scheduling period. It is assumed that time interval of two successive requirements of a machine by parts are exponentially distributed. From the viewpoint of a machine, it is assumed that an interval of two successive requirements during the scheduling period, and the processing times of the parts in the machine are generally distributed. Therefore, the p_i's, the probabilities that machine i can not accommodate a part, can be calculated by M/G/1 queuing theory. If N, the number of parts waiting in front of machine i, is greater than the local buffer capacity of machine i, these parts must be sent to the central buffer.

From the above reasoning, p_i is determined as follows:

$$p_i = \Pr\{N > 1\} = 1 - \sum_{i=0}^{1} \Pr\{N=i\}$$

where the value of $\Pr\{N=i\}$ is given in Gross and Harris(pp.252-264,1985).

임준묵(林俊默)
1988년 서울대학교 산업공학과 학사
1990년 한국과학기술원산업공학과 석사
1994년 한국과학기술원산업공학과 박사
현재 강릉대학교 산업공학과 조교수
관심분야: 공장자동화, 물류관리, 생산시스템설계, 시뮬레이션

황 학(黃 鶴)
1975년 미네소타 주립대학 산업공학과 공학박사
현재 한국과학기술원 산업공학과 교수
관심분야: 공장자동화, 물류관리, 시설계획 및 설계, 재고 및 생산관리, 작업관리