

# 기계 배치 문제의 온라인 알고리즘

왕 지 남\*

## An On-line Algorithm for Machine Layout Problem

Gi-Nam Wang\*

### ABSTRACT

This paper covers algorithms to determine a machine assignment strategy to locations on a single straight track by minimizing the total backtrack distance. Three different algorithms are presented: an efficient heuristic procedure, the branch-and-bound algorithm, and the neural network approach. Simulation results show that the proposed algorithms have potential power to design an on-line optimizer.

**Key Words** : 역류 거리(Backtrack Distance), Heuristic Procedure, Branch-and-Bound, Neural Network

### 1. 서 론

유연 생산 시스템에서는 하나의 작업 라인에서 여러 개의 다른 작업들이 공유하게 되며 인라인 형태에서 작업장은 직선 형태로 배열되는 것이 일반적이다. 본 논문은 생산공정의 기계 배치 문제로 작업장에서의 작업의 역류 거리를 최소화 하는 기계 배치 문제를 다룬다. 기계 배치는 양 방향의 흐름을 갖는 직선 형태의 인라인 FMS 배치로 여러 개의 주어진 작업이 각각의 공정 순서에 따라 기계 배치를 순 방향 또는 역 방향으로 통과하게 된다. 역 방향 통과는 순 방향 통과와는 달리 공정상 역류 거리를 발생할 수 있으므로 역 방향으로 통과하는 전체 역류 거리를 최소화하는 기계 배치 문제를 다룬다. 역류 거리를 최소화 하는 문제의 필요성은

만약에 생산하는 제품의 종류가 단일 품이라면 인라인 형태를 제품의 공정 순서에 맞게 기계를 배치하면 물류의 방향은 항상 왼쪽에서 오른쪽으로 진행되어 역류 거리가 없을 것이다. 그러나 생산하는 제품의 종류가 다양하고 공정의 순서가 서로 다른 경우 역류가 발생될 수 있으며 역류는 물류의 흐름을 역행하게 되어 제조공정의 복잡성을 야기시킬 것이다.

본 논문에서 다루는 기계 배치의 한 경우가 아래 Fig. 1에 나타나 있다.<sup>(5)</sup> 기계 배치  $A=(3, 5, 2, 1, 4)$ 에 대한 역류 거리를 그림으로 나타낸 것이다.

즉  $A=(3, 5, 2, 1, 4)$ 는 기계3을 첫번째 위치에, 기계4를 마지막 위치에 배치하는 경우를 나타낸다. 이 경우 공정 순서가 기계들의 사용율 (2, 1, 3, 4, 5)의 순서로 요구되면 이때의 역류 거리는 순서 (2, 1)에서 0 unit,

\* 아주 대학교 산업공학과(정회원)

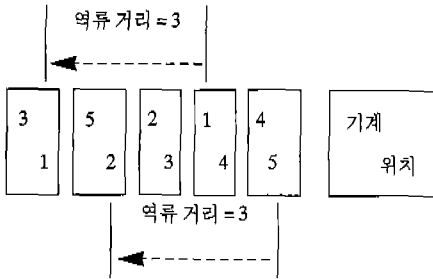


Fig. 1 기계 배치와 역류의 예

순서 (1, 3)에서 3 unit, 순서 (3, 4)에서 0 unit, 순서 (4, 5)에서 3 unit만큼 각각 역류 거리가 발생되어 총 6 unit의 역류 거리가 발생 한다. 상기 문제는 QAP(Quadratic Assignment Problem)으로 표시되며  $X_{ik}$ 를 기계 i를 k위치에 배치하는 결정 변수로 두면 다음과 같이 수식화 된다.

$$\begin{aligned} \text{Minimize } f(x) &= \sum_i \sum_j \sum_k \sum_h C_{ikjh} X_{ik} X_{jh} \\ \text{st } \sum_i X_{ik} &= 1 \quad \text{단 } k = 1, 2, 3, 4, \dots, M \\ \sum_j X_{jh} &= 1 \quad \text{단 } h = 1, 2, 3, 4, \dots, M \\ X_{ik} &= 1 \quad \text{기계 } i \text{가 } k \text{ 위치에 있을 때} \\ X_{ik} &= 0 \quad \text{otherwise} \end{aligned}$$

$C_{ikjh}$ 는 i기계가 k위치에 있고 j기계가 h위치에 있을 때 발생하는 역류 거리를 나타내며  $X_{ik}$ 는 모든 i와 k에 대하여 0 혹은 1의 값을 갖는다. 각 작업장들에게 최적의 기계 배치를 찾아 내는 방법은 많은 계산량을 요하는 NP-Complete 문제로 본 문제와 유사한 문제인 QAP에 대하여 많은 연구가 있었다. 이미 존재하는 방법들 (6, 7, 8)은 문제 크기가 작은 경우에 대하여도 최적 해를 찾아 내는 체계적인 방법이 제시되지 않았으며 큰 문제에 대하여는 최적 해를 찾아 내는데 많은 어려움이 있었다. 이러한 문제를 극복하기 위하여 많은 경험적 방법(Heuristic Procedure) (5, 7, 8)들이 제시되었다. 또한 신경망을 이용하여 계산 속도와 해의 정확도를 향상 시키는 노력도 있었다.<sup>(4, 11)</sup>

본 논문에서는 이미 발표된 경험적 방법 (5)과 동일한 문제에 대하여 다음과 같은 세 가지의 방법을 제시한다. 첫째로, 효율적인 경험적 방법을 개발하고, 둘째로 최적 해를 보장하는 Branch-and-Bound방법과 셋

째로, 신경망을 이용하여 학습을 통하여 실시간으로 기계 배치의 문제를 다루는 방법을 제시하며 최적화 기법과 신경망 기법을 통합하는 연구에 대하여 살펴 보고자 한다. 이상 제시되는 방법은 아직 연구되지 않았던 것으로 특히 학습을 통한 신경망 방법은 QAP의 해를 주어진 문제로 부터 입출력(Input-Output)의 Mapping 관계로 나타내는 접근법으로 처음 제시되는 방법으로 사료 된다.

## 2. 제안된 방법들

### 1) 경험적 방법

이 방법은 이미 발표된 경험적 방법에 초기 해를 구하는 방법을 보완한 것이다. 기본적인 절차인 역류 거리를 계산하는 형태는 이전 연구 논문 (5)과 유사한 것으로 다음과 같이 설명된다.

- 역류 거리 계산 절차

만약 4개의 기계가 배치된 곳에 6개의 작업( $J_1, J_2, \dots, J_6$ )이 진행되고 각 작업들이 다음과 같은 순서로 기계가 사용된다고 하자. 주어진 작업 군으로부터 기계간 작업운반 빈도를 나타내는 R(Relation)행렬을 다음과 같이 구한다. R행렬의 첫번째 열의 마지막 행인 요소( $r_{14} = 3$ )는 기계 1-4간의 작업운반이 작업 군  $J_1, J_2, J_3$ 에서 발생하고 총 3번 나타나므로  $r_{14} = 3$ 이다.

작업	기계 순서
$J_1$	2-3- <u>1</u> -4
$J_2$	2- <u>1</u> -4
$J_3$	<u>1</u> -4-3-2
$J_4$	3-4-2
$J_5$	4-1-3
$J_6$	1-3-2-4

$$R = [r_{ij}] = \begin{bmatrix} 0 & 0 & 2 & 3 \\ 1 & 0 & 1 & 1 \\ 1 & 2 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

R행렬은 주어진 작업 군으로 부터 결정 되어지며 이는 각각의 문제에서 주어지는 작업의 순서를 나타내는 정보라 할 수 있다. R행렬과 관련하여 B(Backtrack)행렬은 결정해야 하는 결정 변수로 부터 얻어지며 기계 배치의 매정 벡터가  $A=(4, 3, 2, 1)$ 이면 즉 기계4를 첫번

제 위치에 기계3을 두번째 위치에 기계2를 세번째 위치에 기계1을 마지막 위치에 배정한 경우로 이때의 역류 행렬은 다음과 같이 구하여 진다.

$$B = [b_{ij}] = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$b_{ij} = \begin{cases} k-h & \text{if } h < k \\ 0 & \text{otherwise} \end{cases}$$

$b_{ij}$ 는 B행렬의 i번째 열과 j번째 행의 원소를 나타내며 역류 행렬의 첫번째 열의 마지막 행인 요소( $b_{14}=3$ )는 기계 배치  $A=(4,3,2,1)$ 로부터 1-4간의 역류 거리가 3임을 나타낸다. R행렬은 주어진 문제에서 확정되는 반면 역류 행렬은 기계 배치로 부터 정하여 진다. 그러므로 주어진 기계 배치  $A=(4,3,2,1)$ 에 대한 전체 역류 거리는 다음과 같이 계산된다.

	열의 합	
$D = [r_{ij}xb_{ij}] = \begin{bmatrix} 0 & 0 & 4 & 9 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$	13 3 1 0 17	

총 역류 거리는 D(Distance)행렬의 모든 요소의 합계 (17)로 표시된다. 이는 발생하는 역류 거리  $b_{ij}$ 를 발생 빈도수  $r_{ij}$ 를 고려하여 곱한후 모든 발생 가능한 경우를 합한 것이다. 그러므로 위의 기계 배치 문제는 주어진 작업 군에서 얻어진 R행렬로부터 역류 거리를 최소화 하는 기계 배치 벡터를 혹은 역류 행렬을 구하는 것이다.

- 제안된 휴리스틱 방법

절차 1.

경험적 방법은 주어진 R행렬을 이용하여 기계의 운반횟수가 가장 많은 기계 순서대로 앞에서부터 배치함으로 초기 해를 구한다.

열의 합

$R = [r_{ij}] = \begin{bmatrix} 0 & 0 & 2 & 3 \\ 1 & 0 & 1 & 1 \\ 1 & 2 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$	5 3 4 3
---	------------------

즉 위의 문제에서 첫번째 열의 합이 5로 가장 크기 때문에 기계1을 첫번째 위치에 그리고 세번째 열의 합이 4로 두번째 크기 때문에 기계3을 두번째 위치에 그리고 두번째와 네번째 열의 합이 각각 3으로 가장 작으므로 기계2와 기계4중 임의로 선정하여 세번째와 네번째에 각각 배치한다. 기계운반회수가 많은 기계를 앞에 배치함으로써 운반이 많은 기계의 역류 거리를 최소화하도록 기계를 배치하고자 하는 것이 기본 논리이다. 초기의 기계 배치 ( $A_0$ )와 해당 역류 행렬 ( $B_0$ )은 다음과 같다.

$A_{0_0} = (1,3,2,4)$	
$B_0 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 3 & 1 & 2 & 0 \end{bmatrix}$	열의 합 0 3 1 6

혹은

$A_{0_0} = (1,3,4,2)$	
$B_0 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 3 & 0 & 2 & 1 \\ 1 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 \end{bmatrix}$	열의 합 0 6 1 3

초기배치 ( $A_0 = (1,3,2,4)$ )에 대한 역류 거리 행렬 ( $D_{A_0}$ )과 총 역류 거리 ( $TB(A_0)$ )는 다음과 같다.

	열의 합
$D_{A_0} = [r_{ij}xb_{ij}] = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 2 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 3 & 1 & 2 & 0 \end{bmatrix}$	0 3 1 6
$TB(A_{0_0}) = 0 + 3 + 1 + 6 = 10$	

절차 2.

기계 배치 (1,3,2,4)에서 기계4가 역류 거리 6unit로 가장 많이 발생하므로 기계4를 가능한 다른 위치로 바꾸어 보며 그때의 역류 거리는 다음과 같다.

$A_{1_1} = (4,1,3,2)$	$TB(A_{1_1}) = 12$
$A_{1_2} = (1,4,3,2)$	$TB(A_{1_2}) = 10$
$A_{1_3} = (1,3,4,2)$	$TB(A_{1_3}) = 10$

**절차 3.**

절차 2에서 제시한 기계 배치는 초기배치보다 역류 거리를 감소시키지 못하므로 두번째 역류 거리를 많이 발생시키고 있는 기계2(3unit)를 다른 위치로 바꾸어 보며 그때 역류 거리는 다음과 같다.

$$\begin{aligned} A1_1 &= (2,1,3,4) \quad TB(A1_1) = 11 \\ A1_2 &= (1,2,3,4) \quad TB(A1_2) = 11 \\ A1_3 &= (1,3,4,2) \quad TB(A1_3) = 10 \end{aligned}$$

**절차 4.**

절차 3에서 제시한 기계 배치는 초기의 배치보다 역류 거리를 감소시키지 못하므로 세번째 역류 거리를 많이 발생시키고 있는 기계3(1 unit)을 다른 위치로 바꾸어 보며 그때 역류 거리는 다음과 같다.

$$\begin{aligned} A1_1 &= (3,1,2,4) \quad TB(A1_1) = 11 \\ A1_2 &= (1,2,3,4) \quad TB(A1_2) = 11: \text{절차 3에서 고려함.} \\ A1_3 &= (1,2,4,3) \quad TB(A1_3) = 10 \end{aligned}$$

어떠한 경우도 초기 해보다 역류 거리를 감소 시키지 못하므로 휴리스틱의 탐색 절차가 종료 되며 초기 해가 제시되는 기계 배치의 해가 된다.

**2) Branch-and-Bound의 최적화 방법**

일반적으로 상기 문제는 최적 해를 얻는데 NP-Complete한 문제로 많은 계산 과정을 거쳐야 하며 완전 나열외에 최적 해를 구하는 방법은 제시되지 않은 것으로 사료된다. 본 논문에서는 Branch-and-Bound 알고리즘을 이용한 최적 해를 구하는 방법을 제시 하고자 한다. 해법은 관계 행렬과 역류 행렬의 특성을 이용하여 부분적으로 배정 벡터를 생성시키면서 하한 값(Lower Bound) 계산에 근거한다.

**부분적 배정 벡터와 부분적 역류 행렬**

부분적 배정 벡터는 Branch-and-Bound에서 체계적으로 기계 배치를 탐색하는 과정에서 생성 되며 부분적 배정 벡터에 해당하는 부분적 역류 행렬을 구할 수 있다.  $A=(3, x, x, x, 4)$ 는 부분적 배정 벡터를 표시한 것이며 기계3을 첫번째 위치에 기계4를 마지막 위치에 배정하고 나머지 기계들은 정하여 지지 않은 것을 의미한다. 부분적 기계배정에 대한 부분적 역류 행렬을 아래와 같이 계산할 수 있다. 만약 기계1이 위치4에 배정되면,  $A=(3, x, x, 1, 4)$ , 부분적 역류 행렬의 미정된 요

소는 줄어 들며 다음과 같이 계산 된다.

$$B = \begin{bmatrix} 0 & x & 3 & 0 & 0 \\ x & 0 & x & 0 & x \\ 0 & 0 & 0 & 0 & 0 \\ 1 & x & 4 & 0 & x \\ x & x & x & 0 & 0 \end{bmatrix}$$

x는 알려 지지 않은 수치를 표시 한다. 상기의 예제는 부분배정으로 부분적 역류계산이 가능함을 보여 주고 있다. 또한 부분적 기계배정 댓 수에 반응하여 부분적 역류 거리를 알아낼 수 있음을 나타낸다. 아래에 앞으로 제시할 해법절차에 필요한 역류 행렬의 특성을 설명 한다.

**<Property 1>**

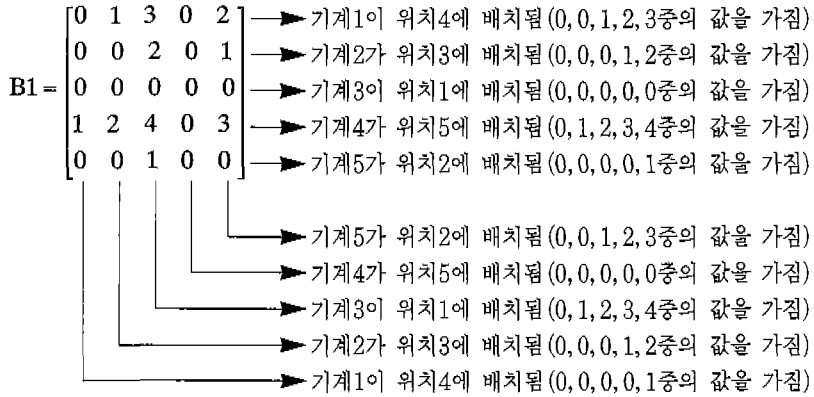
기계의 부분배정이 이루어지면 부분적 역류 행렬을 구할 수 있다. 또한 알려진 배정 갯수에 비례하여 역류 행렬의 알려지는 요소 갯수가 결정된다.

**<Property 2>**

m개의 기계가 배정되면 mxm 역류 행렬이 유일한 형태로 존재하게 된다. 기계 배치  $A1=(3, 5, 2, 1, 4)$ 에 대한 역류 거리 행렬은 다음과 같이 유일하게 존재 한다. 이는 주어진 기계 배치의 크기가 정해지면 정해진 m에 대하여 전체 역류 행렬의 가능한 요소의 값들은 기계 배치에 따라 위치만 달라지고 동일한 요소들의 조합으로 구성된다.

역류 행렬의 요소들의 수치들은 각 기계의 위치가 정해지면 해당열과 행들의 가능한 값들을 알 수가 있다. 그러나 각 요소들의 정확한 위치는 배치벡터에 따라 결정된다. 즉 총 N의 위치 중 기계M이 K번째 위치에 배치되면 역류 행렬의 열 M과 행 M의 요소들은 다음과 같이 결정된다. 열 M의 요소들  $(F(K-1), F(K-2), \dots, F(K-N))$  중에서 선택 되고 행 M의 요소들은  $(F(N-K), F(N-(K+1)), F(N-(K+2)), \dots, F(N-(K+N-1)))$  중에서 선택되어 지며 함수  $F(x)$ 는 다음과 같다.

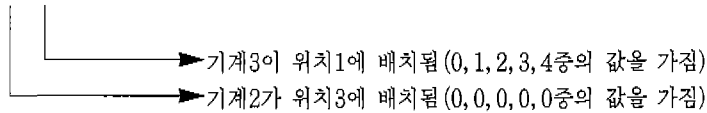
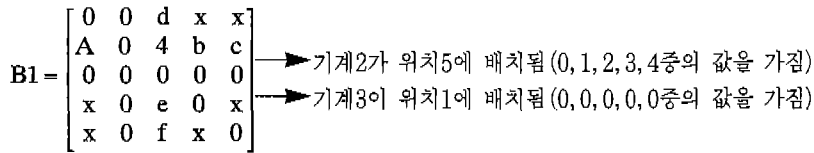
$$F[x] = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$



〈Property3〉

기계의 부분배정에 따라 배정된 기계에 의한 역류 거리의 하한 값을 구할 수 있다. 부분배정 A1=(3, x, x, x, 2)에 대한 역류 행렬은 기계 배치와 역류 행렬의 1대

1 대응과 유일성에 의하여 기계의 부분배정으로 부터 역류 행렬의 부분배정을 계산할 수 있다. 또한 역류 행렬의 부분배정은 전체 역류 행렬의 유일성으로 부터 가능한 역류 거리의 최소값을 구할 수 있다.



위에서 역류 행렬의 두번째 열의 요소 중 (a, b, c)의 값은 (1, 2, 3)의 요소 중 하나씩 배정되어 구성되며 역류 행렬의 세번째 행인 (d, e, f)들 역시 (1, 2, 3)중에서 하나씩 배정되어 구성되는 것을 알수 있다.

$$R = \begin{bmatrix} 0 & 0 & 2 & 3 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 2 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 3 & 1 & 2 & 1 & 0 \end{bmatrix}$$

만약 작업 군으로 부터 R행렬이 위와 같이 주어 졌다면 A1에 대한 역류 거리의 하한 값은 다음과 같다.

$$LB(A1) = (B1\text{요소 중 확정된 요소들}) \times (R\text{요소 중 해당 요소들}) + (B1\text{에서 } a, b, c\text{의 증가 순}) \times (\text{해당 } R\text{요소의 감소 순}) + (B1\text{에서 } d, e, f\text{의 증가 순}) \times (\text{해당 } R\text{요소의 감소 순}) + (B1\text{에서 나머지 } (x)\text{ 요소의 증가 순}) \times (\text{해당 } R\text{요}$$

소의 감소 순)

$$= 4 \times 1 + (1, 2, 3) \times (1, 1, 0)^t + (1, 2, 3) \times (4, 3, 1)^t + (0, 0, 0, 1, 1, 2) \times (3, 3, 1, 1, 1, 0)^t = 22$$

주어진 작업 군에 의하여 얻어진 R행렬이 다음과 같을 때 위의 절차를 설명하면 다음과 같다.

열의 합

$$R = \begin{bmatrix} 0 & 0 & 1 & 3 & 2 \\ 4 & 0 & 1 & 2 & 1 \\ 2 & 3 & 0 & 1 & 2 \\ 0 & 2 & 1 & 0 & 2 \\ 3 & 2 & 1 & 1 & 0 \\ 9 & 7 & 4 & 7 & 7 \end{bmatrix} \begin{matrix} 6 \\ 8 \\ 8 \\ 5 \\ 7 \end{matrix}$$

행의 합

Step 1. 초기 해를 구한다.

상기의 행렬에서 행의 합, 열의 합을 통해서 가장 큰

값은 행의 합인 9로 기계1에 해당한다. 따라서 가장 마지막에 기계1을 배정한다. 나머지 열의 합도 크기가 동일하므로 행의 합의 크기에 따라서 나머지 위치를 배정하면  $A^*=(3, 2, 5, 4, 1)$  이 된다.

열의 합

$$B = \begin{bmatrix} 0 & 3 & 4 & 1 & 2 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 3 & 0 & 1 \\ 0 & 1 & 2 & 0 & 0 \end{bmatrix} \begin{matrix} 10 \\ 1 \\ 0 \\ 6 \\ 3 \end{matrix}$$

행의 합  $RB = 4 + 3 + 4 + 1 + 4 + 3 + 2 + 2 + 2 = 25$

Step 2 Branching과 Fathoming과정

- Iteration 1:  $A=(1, x, x, x, 5)$

$$R = \begin{bmatrix} 0 & 0 & 1 & 3 & 2 \\ 4 & 0 & 1 & 2 & 1 \\ 2 & 3 & 0 & 1 & 2 \\ 0 & 2 & 1 & 0 & 2 \\ 3 & 2 & 1 & 1 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ x & 0 & x & x & 0 \\ x & x & 0 & x & 0 \\ x & x & x & 0 & 0 \\ 4 & x & x & x & 0 \end{bmatrix} \begin{matrix} \rightarrow 4+3+2+1=10 \\ \text{4번째 열의 합} \\ \rightarrow 4+3+2+1=10 \text{(첫번째 행의 합)} \end{matrix}$$

$$LB(A) = (3 * 4) + (0, 2, 4) * (3, 2, 1) + (1, 1, 2) * (3, 2, 1) + (1, 1, 1, 2, 2, 3) * (2, 1, 1, 0, 0, 0)$$

- Iteration 2:  $A=(1, x, x, x, 4)$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ x & 0 & x & x & 0 \\ x & x & 0 & x & 0 \\ 4 & x & x & 0 & 0 \\ x & x & x & x & 0 \end{bmatrix} \begin{matrix} \rightarrow 1+2+3+4=10 \\ \rightarrow 4+3+2+1=10 \text{(첫번째 행의 합)} \end{matrix}$$

Step 3 종료 검사

- 더 이상 Branching이 없으면 종료한다. 상기의 계산 과정을 거치면 최적 해는  $A^*=(3, 2, 5, 1, 4)$ 이며 이때  $TB=24$ 이다. 계략적 계산 과정을 보면 다음과 같

Table 1 Branch-and-Bound 계산 결과

Iteration	a	Lower bound	Fathom	Current optimal
Initial sol	(3,2,5,4,1)	25		(3,2,5,4,1)
1	(1,x,x,x,5)	31	Fathomed	(3,2,5,4,1)
2	(1,x,x,x,4)	29	Fathomed	(3,2,5,4,1)
3	(1,x,x,x,3)	33	Fathomed	(3,2,5,4,1)
...	...	...	...	...
18	(5,x,x,x,3)	29	Fathomed	(3,2,5,1,4)
19	(5,x,x,x,2)	29	Fathomed	(3,2,5,1,4)
20	(5,x,x,x,1)	26	Fathomed	(3,2,5,1,4)

다. Table 1은 상기문제에 대한 Branch-and-Bound의 각 절차의 계산 과정을 도표로 나타내었다. 제안된 Branch-and-Bound를 이용한 경우 모든 가지수 계산 ( $5!=120$ )보다 최적 해를 구하는 과정에서 계산 횟수를 줄일 수 있었다.

Fig. 2는 기본적인 기계 배치 문제에 있어서 Branch-and-Bound과정을 보여 주고 있다.

Branch-and-Bound과정은 계층적인 나무 구조로 체계적인 부분적 기계 배치를 생성하면서 최적 해를 찾아가는 방법으로 문제의 효율성은 각 부분적 기계 배치에 대하여 하한 값의 밀착성(Tightness)에 따라 계산량을 줄일 수 있다.

3) 신경망에 의한 방법

최근에 신경망을 최적화 문제에 적용하여 해의 정확

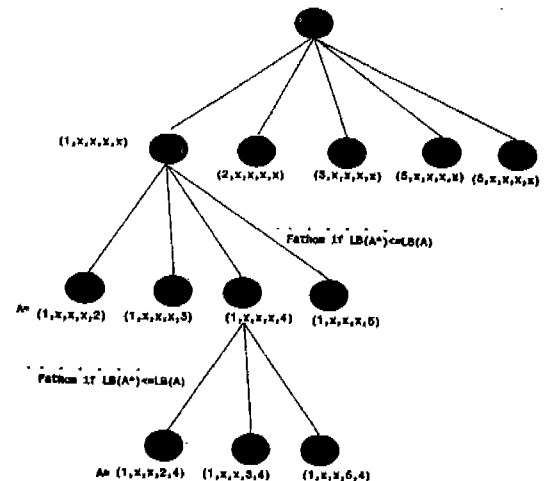


Fig. 2 Branch-and-Bound 절차 과정

도와 계산량을 향상시키려는 연구가 많은 관심을 끌고 있다. 본 문제와 일치하지는 않지만 일반적인 배치 문제에 신경망을 효과적으로 응용한 방법을 최근에 제시하였다.<sup>(11,12)</sup> 이들의 방법은 신경망을 이용 재 모델링한 에너지 함수를 사용하여 해를 구하고 있다. 본 논문에서는 신경망 특성 중 입력과 출력의 Mapping관계를 학습을 통하여 신경망 모델 중 역전파 신경망(Back-propagation)을 이용하여 실시간에 기계 배치 문제를 처리한다는 점에서 이전의 방법들과 다르다고 할 수 있다.

- 입력-출력 Mapping의 적용

신경망에 대한 많은 정의 중 Mapping은 중요한 신경망의 특성이다. 본 문제와 관련시켜 설명하면 입력은 기계 배치의 문제이고 출력은 기계 배치의 해라 볼 수 있다. 본 기계 배치 문제는 다음과 같은 Mapping의 관계로 정의 할 수 있다. 주어진 문제 즉 작업 군의 특성을 나타내는 R행렬로부터 최적의 기계 배치를 나타내는 A 벡터를 구하는 것이다. 그러므로 본 연구에서 설계된 신경망의 입력은 문제로 부터 얻어진 R행렬의 요소를 신경망의 입력으로 기계 배치를 나타내는 A 벡터를 출력으로 설계할 수 있다.

신경망의 입력과 출력을 결정하는 것은 신경망의 구조(Architecture)설계의 한 부분으로 전체적인 구조는 입출력의 수와 종류 결정, 입출력을 연결하는 중간층의 수와 각층을 구성하는 노드(Node)수, 노드 안의 통과함수(Transfer Function)를 결정하는 일이다. 본 논문에서 사용된 신경망의 구조를 Fig. 3에서 나타내었다.

- 학습 알고리즘

기계 배치 문제에서 학습은 R행렬로부터 최적의 기계 배치 벡터를 생성하는 중간층의 연결부(Connection Weight)를 찾아 내는 과정으로 주어진 학습을 위한 학

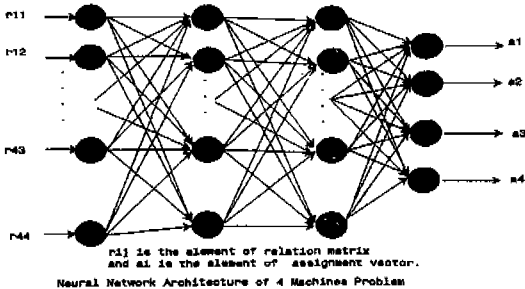


Fig. 3 기계 배치의 신경망 구조

습 데이터군(Training Set)을 이용하여 학습한다. 학습을 위한 데이터군은 여러경우의 작업 군을 표시 하는 R행렬들과 각각의 경우의 최적 기계 배치 벡터들을 이용하여 주어진 R행렬의 입력에 대하여 최적 기계 배치 벡터를 출력으로 Mapping할 수 있는 연결부를 학습을 통하여 정한 후 실제의 기계 배치를 구하는데 이미 학습을 통하여 얻어진 입력-출력 Mapping구조를 이용하여 실시간 기계 배치 해를 찾아 내는 방법이다. 이러한 학습방법은 오프라인 학습으로 잘 알려진 역전파 신경망의 학습 알고리즘으로 다음과 같이 설명된다.

절차 1: 신경망 구조 결정

신경망의 입출력 구조를 선정하고 각 노드에서 사용되는 변환 함수를 결정하고 연결부에 초기값을 Random으로 부여한다. 사용된 변환 함수는 Sigmoid 함수이다.

$$f(x) = 1 / (1 + e^{-(x-\theta)})$$

절차 2: 학습 데이터군 설정

입출력 구조에 형태에 합당한 학습 데이터군을 선정한다. 학습 데이터군은 신경망의 전체 성능에 영향을 미치므로 전체 문제 영역을 대표할 수 있도록 선정해야 한다. 신경망의 일반화 능력(Generalization Capability)으로 평가되는 학습 데이터와 전체 성능은 학습 데이터가 편중함에 따라 일반화 능력은 감소할 것이다. 기존의 연구(12)에서 신경망의 일반화 성능은 전형적인 대표 군(Typical Set)과 대표에서 벗어난 외곽 군(Critical Set)의 조합으로 학습 데이터를 선정 하였을 때 전체 성능이 향상됨을 보여 주었다.

절차 3: 학습과정

모든 학습 데이터군의 입력에 대하여 신경망의 출력과 학습 데이터군의 출력과의 차이를 최소화 하도록 연결부의 값을 연속적으로 수정한다.

- 3.1 학습 데이터 군을 이용 신경망 출력과 목표 출력을 계산.

$(x_1, x_2, \dots, x_N)$ 은 학습 데이터군의 입력,  $(d_1, d_2, \dots, d_M)$ 은 학습 데이터군에 나타난 목표 출력,  $(y_1, y_2, \dots, y_M)$ 은 신경망의 출력을 표시한다.

- 3.2 신경망 연결부를 수정. 신경망 연결부를 학습 하는 학습 알고리즘은 잘 알려진 Steepest Descent Search방법으로 반복적인 탐색으로 신경망 연결부를 수정한다.

$$W_{ij}(t+1) = W_{ij}(t) + \alpha_1 \beta_j x_j' + \alpha_2 (W_{ij}(t) - W_{ij}(t-1))$$

$$0 < \alpha_1 < 1, 0 < \alpha_2 < 1,$$

만약 노드j가 마지막 출력 층이면  $\beta_j = y_i(1 - y_i)(d_j - y_i)$ , 만약 노드j가 중간층이면  $\beta_j = x_j'(1 - x_j') \sum_k \beta_k$   $W_{jk}$ ,  $W_{ij}$ 는 신경망의 입력 노드 i와 출력노드 j를 연결 하는 연결부를 나타내며  $\alpha_1$ 은 학습 변수,  $\alpha_2$ 는 Moment변수이다.

- 3.3 모든 학습 데이터군에 대하여 신경망 출력과 목표 출력의 차이가 정해진 기준치 이하가 될 때까지 3.1과 3.2를 반복한다. 이 기준치는 신경망의 학습 정도를 나타내는 수치로 학습 시간이 경과함에 따라 신경망 출력과 목표 출력의 차이의 전체 어라가 일반적으로 감소하게 되고 어느 시점에 이르면 일정한 값에 근사적으로 수렴하게 된다. 수렴된 값을 기준치로 정하여 이후의 학습 효과는 크게 기대할 수 없으므로 학습 중단의 기준치로 정한다.

절차 4: 실제 문제에 적용

학습된 신경망을 학습 데이터군과 다른 입력에 대하여 신경망의 출력과 최적 기계 배치 안을 비교 학습된 신경망의 성능을 평가한다.

3. 실험 및 결과 분석

위에서 제시한 세 가지의 서로 다른 방법에 대한 성능 평가를 위하여 주어진  $m(m=4, 5, \dots, 30)$ 개의 기계에 대하여 50개의 서로 다른 R행렬을 난수로 생성하였다. 난수의 범위는 0에서 부터 20까지로 하여 최대 기계운반회수가 작업 군에서 20번으로 하여 일반화된 넓은 범위로 하였다.

Table 2은 완전 나열에 의한 방법, 이미 발표된 휴리스틱 방법1(휴1) (5), 본 논문에서 제시된 휴리스틱 방법(휴2), 그리고 Branch-and-Bound의 성능들을 계산량과 계산 시간, 그리고 해의 정확도 측면에서 비교 하였다. 계산량이란 각 방법에서 사용된 총 Iteration의 수를 나타내고 정확도란 각 방법에서 얻어진 총 역류 거리를 최적 해에서 얻어진 총 역류 거리로 나눈 값으로 1에 가까울수록 정확도가 높은 것을 의미 한다. Table 2에 나타난 바와 같이 제안된 휴리스틱(휴2)은 기존의 휴리스틱 보다 계산량, 정확도, 그리고 계산 시간에서 모두 성능이 향상되었으며 또한 제시된 Branch-and-Bound 역시 총 나열보다 훨씬 적은 계산량으로 최적 해를 구할 수 있었다. 최적 해를 구하는 Branch-and-Bound는 휴리스틱 보다는 최적 해를 얻

Table 2 기계 배치 알고리즘의 성능비교

m	계산량				정확도			계산 시간(초)		
	나열(m!)	휴1	휴2	BB	휴1	휴2	BB	휴1	휴2	BB
4	24	18.2	15.1	12	1.082	1.056	1	0.000	0.000	0.000
5	120	26.66	23.2	49	1.091	1.067	1	0.000	0.000	0.020
6	720	38.33	34.33	167	1.172	1.079	1	0.000	0.000	0.060
7	5040	51.6	46.8	647	1.192	1.002	1	0.041	0.033	0.220
8	20302	66.6	61.7	2538	1.202	1.012	1	0.074	0.066	1.220
9	182718	99.0	89.0	4534	1.351	1.074	1	0.159	0.100	2.660
10	1827180	143	126	7692	1.368	1.091	1	0.187	0.110	5.600
12	241187760	643.0	519.09	27912	1.395	1.097	1	0.327	0.300	1105.0
15	15!	1285	1021	84353	1.392	1.121	1	0.651	0.431	3421.0
20	20!	3287	2871	931322	1.431	1.154	1	5.782	3.231	87321
30	30!	98212	4211	5324530	1.493	1.234	1	10.323	5.321	432121



Table 3 신경망을 이용한 기계 배치의 성능

m	학 습		시 험	
	최적해 회수율	정확도	최적해 회수율	정확도
4	98%	1.00931	25%	1.247875
6	96.4%	1.01238	16%	1.477293

는데 계산량과 계산 시간이 많이 소요되었지만 기계가 10대인 경우 평균 5.6초안에 최적 해를 구하였다. 참고로 실험에 사용된 컴퓨터는 PC-486(66 MHz)이다.

신경망을 이용한 기계 배치 문제는 우선 기계 대수가 4대와 5대인 경우에 실험 하였으며 50개의 서로 다른 작업 군에서 최적 해를 구하여 R행렬을 차례로 신경망의 입력으로 하고 최적 해의 기계 배치 벡터(A)를 신경망의 출력으로 하여 학습 하였다. 학습된 신경망의 성능을 시험하기 위하여 다시 새로운 100개의 학습에 사용 되지 않은 서로 다른 R행렬을 난수로 만들어 신경망의 입력으로 하여 신경망의 출력과 Branch-and-Bound에서 구한 최적 해와 비교하였다. Table 3는 기계 대수가 4대와 6대인 경우의 신경망의 성능을 제시한 것이다. 최적 해 회수율은 신경망의 출력이 최적 해일 경우를 백분율로 표시한 것으로 학습된 데이터의 경우에는 높은 비율을 나타내었고 학습되지 않은 경우에는 16%와 25%를 보였다. 그러나 정확도 면에서는 비교적 최적 해와 가까운 해를 제시하였다.

신경망을 이용한 경우에는 신경망의 Mapping구조, 그리고 학습 데이터를 연구함으로써 보다 좋은 성능을 제시할 것으로 사료되며 신경망방법과 Branch-and-Bound의 방법 등을 접합하여 보다 효율적인 기계 배치 해를 제시할 것으로 보인다. 신경망의 경우에는 비록 적은 기계 대수의 경우만 실험하였으나 본 연구는 일반적인 기계 대수의 경우에도 확장 가능 하리라고 보며 이에대한 연구가 앞으로 과제라 볼 수 있다.

#### 4. 결 론

본 연구에서 생산공정에 있어서 기계 배치 문제를 휴리스틱방법, 최적 해 절차, 그리고 신경망을 이용하여 온라인으로 해를 구하는 방법을 살펴 보았다. 최적 해를 찾는 Branch-and-Bound방법은 다른 두 방법에 비하여 많은 계산량을 요구하나 최적 해를 구할 수 있다는 점에서 장점을 가지고 있다. 제안된 휴리스틱 방법을 비교적 적은 계산량으로 최적 해에 가까운 해를

구할 수 있었다. 신경망을 이용한 방법은 학습된 신경망의 구조를 이용하여 별도의 특별한 계산 과정 없이 온라인을 해를 얻을 수 있으나 최적 해의 정확도면에서 세 가지 방법중 가장 뒤떨어 졌다. 문제의 상황에 따라 각각의 방법을 선택 하여 이용 할 수 있겠다. 만약 실시간으로 각각 변화하는 상황에서의 기계 배치 해를 구하는 것이 요구 된다면 신경망에 의한 방법도 고려 할 수 있을 것이다. 본 연구에서 제시된 계산 시간의 절약은 문제의 크기가 매우 큰 경우(Large Scale Problem) 현재의 컴퓨터로도 다루기 어려운 실제의 문제가 자주 발생하는 바 큰 문제의 해법을 제시할 수 있다는 면에서 계산 시간의 절약은 의의가 있다고 사료된다.

추후 연구방향으로 본 예는 단일 생산라인으로 단순화한 형태이나 실제 상황에서는 이 보다 복잡한 다중 생산라인 형태의 기계 배치 문제를 고려 할 수 있을 것이다. 또한 단지 역류 거리뿐만 아니라 주어진 작업 군의 총 순 방향과 역 방향의 총 이동 거리를 최소화 하는 문제도 생각 할 수 있겠다. 신경망의 온라인 학습과정을 설계하고 전용 하드웨어를 구현하여 다른 최적화 또는 휴리스틱방법과 연합시켜 온라인 기계 배치 문제의 해를 향상 시키는 방법도 앞으로의 주요한 연구 과제라 하겠다. 작업장의 전체적인 성능을 평가하는 시뮬레이션기법과 신경망을 이용하여 온라인 시스템 설계와 통합시키는 방법도 또 다른 연구 과제라 하겠다.

#### 참고문헌

1. Bharitkar, S., K. Tsuchiya, and Y. Takefuji, "Neural Computing for the Facilities Layout Problem", WCNN '93, Portland, Oregon, July 11-15, Vol.IV, 857-860, 1993.
2. Francis, R. L and White, J. A., Facility Layout and Location:an analysis approach, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1974.
3. Gilmore, P.C., "Optimal and Suboptimal Algorithms for Quadratic Assignment Problems", SIAM Journal on Applied Mathematics, 10.2. pp.305-313, 1962.
4. Gilmore, P.C., 1962, "A Solution to the Module Placement Problem", IBM

- Research Report RC-430, April, 1962.
5. Han, Min-Hong, "Machine Layout Problem in a Production Line", Journal of the Korea Institute of Industrial Engineering, Vol.15, No.1, June, 1989.
  6. Hillier, F. S., "Quantitative Tools for Plant Layout Analysis", The Journal of Industrial Engineering, Vol.15, pp.33-40, 1963.
  7. Heragu, S. S., and Kusiak, A., "Machine Layout in Flexible Manufacturing Systems", Operations Research, pp.258-267, 1988.
  8. Heragu, S. S., and Kusiak, A., "Efficient Models for the Facility Layout Problem", European Journal of Operations Research (Special issue on facility layout, Vol.53, 1991.)
  9. Lawler, E. L., "The Quadratic Assignment Problem", Management Science, Vol.9, pp.586-599, 1963.
  10. Gavett, J. W., And N. V. Plyter, "The Optimal Assignment of Facilities to Location by Branch and Bound", Operations Research, Vol.14, pp.210-232, 1966.
  11. Yip, Percy P. C., and Pao, Yoh-Han, "A New Optimizer for the Facility Layout Problem", Preceeding of 1993 International Joint Conference on Neural Networks, Nagoya Japan, October 25-29, pp.1573-1576, 1993.
  12. M. Wann, T. Hediger and N. N. Greenbaun, "The Influence of Training Sets on Generalization in Feed-forward Neural Networks", International Joint Conference on Neural Networks, Vol.3, pp.137-142, San Diego, California June 17-21, 1990.