

다수요인을 가진 설비배치문제를 위한 모형과 simulated annealing 알고리즘

홍관수*

A Model and Simulated Annealing Algorithm for the Multi-factor Plant Layout Problem

Kwan Soo Hong*

Abstract

This paper presents a model and algorithm for solving the multi-factor plant layout problem. The model can incorporate more than two factors that may be either quantitative or qualitative. The algorithm is based on simulated annealing, which has been successfully applied for the solution of combinatorial problems. A set of problems previously used by various authors is solved to demonstrate the effectiveness of the proposed methods. The results indicate that the proposed methods can yield good quality for each of eleven test problems.

1. 서론

공장의 효율성은 필요한 설비나 기계의 배치 형태에 따라 크게 영향을 받는다. 적정장소에 배치되지 못한 기계나 설비는 공정재고의 증가, 비효율적인 기계조정작업, 긴 대기행렬, 자재취급시스템의 과부하 등 여러 가지 문제점을

초래한다[12]. 따라서 배치문제는 공장에서 발생하는 중요한 의사결정문제 중의 하나이다. 이러한 배치문제는 이차할당문제(Quadratic Assignment Problem : QAP)로 모형화할 수 있고 QAP를 해결하기 위한 알고리즘은 크게 최적화 알고리즘과 휴리스틱 알고리즘으로 분류할 수 있다.

최적화 알고리즘은 조합적인 성격(combina-

* 계명대학교 경영학과

torial nature) 때문에 배치하여야 할 설비의 수가 15개 이상인 경우에는 경험적으로 보아 최적해를 구하는 것이 불가능하다[4]. 따라서 배치문제를 해결하기 위한 여러 가지 휴리스틱 알고리즘이 개발되었다. 이러한 휴리스틱 알고리즘은 초기배치안작성(construction)과 개선해 도출(improvement) 알고리즘으로 분류할 수 있다[24].

초기배치안작성 알고리즘은 완전한 배치안 작성될 때까지 한번에 하나의 설비를 하나의 지역에 배치한다[9,14,23,31]. 반면 개선해 도출 알고리즘은 주어진 초기배치안을 기초로 하여 모든 설비상호간의 비교를 통하여 해를 가장 많이 개선할 수 있는 설비를 찾아 교환하는 과정을 더 나은 대안이 없을 때까지 계속적으로 반복하여 개선해를 구한다[1,3,15,16,27,28].

그러나 이러한 알고리즘들의 최종배치안은 초기배치안에 의해 크게 영향을 받는 단점을 가지고 있다. 즉, 이들 알고리즘은 목적함수를 개선할 수 있는 대안만을 받아들이는 이른바 greedy 알고리즘에 기초로 하기 때문에 목적함수가 하나 이상의 국부최소점을 가지고 있는 경우에는 초기배치안의 설정에 따라 최종배치안은 주어진 초기배치안과 근접한 국부최소점에 수렴하게 되어 일반적으로 전체 최소점을 찾지 못하게 된다.

이러한 형태의 알고리즘들은 또한 크게 두 가지 접근방식이 있다. 하나는 양적요인인 설비들간의 자재취급비용을 최소화하는 방식이고 다른 하나는 Muther[26]가 제시한 질적요인인 설비들간의 근접도(closeness rating)를 최대화하는 방법이다. 두 가지 방식 서로 장단점을 가지고 있지만 이러한 두 가지 요인들은 동시에 고려되어야만 바람직하다. Rosenblatt[30]가 두 가지 접근방식을 통합한 모형을 제시한

이래로 여러 학자들에 의해 새로운 접근방식이 제시되었다[8,10,13,32].

그러나 다수의 요인들을 동시에 고려한 배치비용의 계산은 학자들마다 다르게 제시하였다. 이러한 배치비용의 계산방법에 따라 각 요인의 상대적 중요도를 나타내는 가중치의 반영정도가 달라진다. 기존의 배치비용의 계산은 각 요인이 배치안에 동등한 영향을 미칠 수 있도록 원래의 설비간의 관계값을 표준화하지 않은 상태에서 각 요인의 가중치를 반영하였기 때문에 각 요인의 가중치가 올바르게 반영되지 못하는 단점을 가지고 있다. 따라서 두가지 이상의 요인들의 가중치를 효율적으로 반영할 수 있는 모형을 제시하는 것이 본 연구의 첫번째 목적이다.

기존의 다수요인을 고려한 모형의 해의 도출은 CRAFT의 방법과 유사한 greedy 알고리즘을 사용하였다[8,10,13,32]. 이러한 반복적 개선법의 단점을 극복하기 위하여 비용함수를 증가시키는 대안들도 받아들임으로써 국부최소해에 수렴되는 것을 방지할 수 있는 simulated annealing(SA) 알고리즘을 자재취급비용만을 고려한 설비배치문제에 적용하여 기존의 greedy 알고리즘과 비교하여 SA의 결과가 보다 우수하다는 것을 제시하였다[17,20,34].

그러나 이들의 SA 알고리즘은 단일요인만을 고려하였고 또한 다양한 유형의 배치문제를 대상으로 실험을 하지 않은 결과이다. 따라서 본 연구의 두번째 목적은 단일요인 또는 다수요인을 가진 다양한 배치문제에 효율적인 SA 알고리즘을 개발하는 것이다. 본 연구에서 제시한 모형과 알고리즘의 효율성을 측정하기 위해 기존의 연구들과 비교하고자 한다.

2. 다수요인을 고려한 모형

다수요인을 고려한 일반적인 이차할당모형은 다음과 같다[32].

$$\text{최소화 } Z = \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n A_{ijk} X_{ij} X_{ki}$$

제약조건

$$\sum_{j=1}^n X_{ij} = 1 (j=1, 2, \dots, n)$$

$$\sum_{i=1}^n X_{ij} = 1 (i=1, 2, \dots, n)$$

$$X_{ij} = 0 \text{ 혹은 } 1 \text{ for all } i, j$$

단

$$X_{ij} = \begin{cases} 1 & (\text{설비 } i \text{가 지역 } j \text{에 할당된 경우}) \\ 0 & (\text{설비 } i \text{가 지역 } j \text{에 할당되지 않는 경우}) \end{cases}$$

$$A_{ijk} = \text{설비 } i \text{를 지역 } j \text{에 그리고 설비 } k \text{를 지역 } i \text{에 할당하는데 따른 배치비용}$$

설비들간의 배치비용을 나타내는 A_{ijk} 은 학자들마다 다르게 정의하였다. Rosenblatt[30]는 $A_{ijk} = \alpha_2 \cdot C_{ijk} - \alpha_1 \cdot R_{ijk}$ 로 정의하였다. 여기서 C_{ijk} 는 설비간의 자재흐름을 나타내는 f_{ik} 와 거리를 나타내는 d_{ji} 의 곱으로 자재취급비용을 나타내고 R_{ijk} 는 설비간의 근접점수를 나타낸다. α_2 와 α_1 은 자재흐름과 근접도 요인에 부여한 가중치를 의미한다. 따라서 그의 모형은 자재취급비용과 근접점수의 가중치화된 차이를 최소화하는 것이다.

Fortenber & Cox[10]는 $A_{ijk} = f_{ik} \cdot d_{ji} \cdot r_{ik}$ 로 표시하여 곱셈모형(multiplicity model)으로 정의하였다. Urban[32]은 위의 모형은 설비간의 흐름이 아주 많고 X 관계에 있는 설비의 경우와, 설비간의 흐름이 매우 적고 A 관계에 있는 경우에는 극단적으로 작은 값을 가지는 모순점

을 지적하였다. 그는 이러한 문제점들을 해결하기 위해 배치비용을 $d_{ji}(f_{ik} + C \cdot r_{ik})$ 로 정의하였다. 여기서 C는 자재흐름 보다 근접도 요인의 중요도를 나타내는 가중치로서 상수이다. 그는 C의 값으로 f_{ik} 의 최대값을 사용하였으나 이는 설비들간의 자재흐름보다는 근접도에 더 높은 가중치를 부여한 결과가 된다.

위의 모형들은 자재흐름과 근접도와 같은 두 가지 요인만을 고려한 모형이나 Harmonosky와 Toth[13]는 두개 이상의 요인을 고려할 수 있는 $A_{ijk} = \sum_{m=1}^l \alpha_m \cdot T_{ikm} \cdot d_{ji}$ 을 제시하였다. 여기서 α_m 은 각 요인의 중요도를 표시하는 가중치이다. T_{ikm} 은 고려하여야 할 요인 m의 설비 i와 k의 관계값을 모든 설비들의 관계값을 합한 값으로 나누어 전체 설비들의 관계값들의 합이 1이 되도록 표준화한 것이다.

그러나 이러한 기존의 배치비용은 각 요인이 배치안에 동등한 영향을 미칠 수 있도록 설비간의 관계값을 표준화하지 않은 상태에서 각 요인의 상대적 중요도를 나타내는 가중치를 반영하였기 때문에 각 요인의 가중치가 올바르게 반영되지 못하였다. 따라서 본 연구에서는 각 요인이 배치안에 동등한 영향을 미칠 수 있도록 각 요인의 설비간의 관계값 중에서 최대값을 1로 만들고 나머지 관계값의 차이의 비율은 그대로 유지할 수 있도록 다음의 식에 의하여 각 요인의 원래값을 표준화하였다.

$$R_{ikm} = C_{ikm} / M_m$$

여기서 R_{ikm} 은 요인 m번째의 설비 i와 k의 표준화된 관계값이고, C_{ikm} 은 요인 m번째의 설비 i와 k의 원래의 관계값을, 그리고 M_m 은 요인 m번째 설비들의 원래의 관계값 중에서 최

대값을 의미한다. 본 연구에서 사용할 모형은 다음과 같다.

$$\text{최소화 } Z = \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \sum_{l=1}^n \sum_{m=1}^l \alpha_m R_{ikm} d_{jl} X_{ij} X_{kl}$$

제약조건

$$\sum_{i=1}^n X_{ij} = 1 (j=1, \dots, n)$$

$$\sum_{j=1}^n X_{ij} = 1 (i=1, \dots, n)$$

$$X_{ij} = 0 \text{ 혹은 } 1 \text{ for all } i, j$$

단

$$X_{ij} = \begin{cases} 1 & (\text{설비 } i \text{가 지역 } j \text{에 할당된 경우}) \\ 0 & (\text{설비 } i \text{가 지역 } j \text{에 할당되지 않는 경우}) \end{cases}$$

α_m = m번째 요인의 중요도를 표시하는 가중치

R_{ikm} = 요인 m번째의 설비 i와 k의 표준화된 관계값

d_{jl} = 지역 j와 l의 거리

t = 고려하여야 할 요인의 수

n = 배치하여야 할 설비의 수

본 연구에서 제시된 모형의 배치비용을 계산하기 위해서는 다음과 같은 과정이 필요하다.

- 1) 계량적분석을 위해 각 요인이 비교가능한 단위를 가질 수 있게 질적요인의 설비간의 관계값을 계량화한다.
- 2) 각 요인이 배치안에 동등하게 영향을 미칠 수 있도록 위에서 설명한 방법으로 표준화한다.
- 3) 각 요인의 상대적 중요도가 배치안에 반영될 수 있도록 각 요인의 가중치를 앞 단계에서 구한 표준화된 값에 곱한다.
- 4) 모든 요인을 하나의 혼합요인(composite factor)으로 만든다. <표 1>과 같은 설비간의 근접도와 자재흐름을 나타내는 두개의 요인만을 가진 단순한 예제를 사용하여 각 단계를 설명하기로 한다.

먼저 각 요인의 설비간의 관계값이 비교가능

<표 1> 예제 문제

	질적요인						양적요인						
	1	2	3	4	5	6	1	2	3	4	5	6	
1		E	O	U	A	I	1		40	60	20	40	40
2	E		E	U	A	U	2	40		40	20	20	80
3	O	E		X	U	X	3	60	40		20	20	60
4	U	U	X		U	U	4	20	20	20		60	20
5	A	A	U	U		A	5	40	20	20	60		100
6	I	U	X	U	A		6	40	80	60	20	100	

한 단위를 가지도록 질적요인을 나타내는 설비간의 근접도를 계량화한다. 여기서는 기존의 연구[13,32]와 같이 A = 4, E = 3, I = 2, O = 1, U = 0, X = -1의 값을 사용하기로 한다. 근접도에서 X 관계에 있는 설비들에 부

(-)의 값을 부여함으로써 이러한 설비간의 인접을 못하게 하는데 큰 효과를 가지는 것으로 나타났다[10,32]. 이와 같이 계량화된 설비간의 관계값은 <표 2>와 같다.

〈표 2〉 계량화된 관계값

질적요인						양적요인							
1	2	3	4	5	6	1	2	3	4	5	6		
1		3	1	0	4	2	1		40	60	20	40	40
2	3		3	0	4	0	2	40		40	20	20	80
3	1	3		-1	0	-1	3	60	40		20	20	60
4	0	0	-1		0	0	4	20	20	20		60	20
5	4	4	0	0		4	5	40	20	20	60		100
6	2	0	-1	0	4		6	40	80	60	20	100	

다음은 각 요인이 배치안에 동등한 영향을 미칠 수 있도록 각 요인의 설비간의 관계값 중에

서 최대값을 1로 만들고 나머지 관계값은 각각의 최대값으로 나누어 〈표 3〉과 같이 작성한다.

〈표 3〉 표준화된 관계값

질적요인						양적요인							
1	2	3	4	5	6	1	2	3	4	5	6		
1		0.75	0.25	0.00	1.00	0.50	1		0.40	0.60	0.20	0.40	0.40
2	0.75		0.75	0.00	1.00	0.00	2	0.40		0.40	0.20	0.20	0.80
3	0.25	0.75		-0.25	0.00	-0.25	3	0.60	0.40		0.20	0.20	0.60
4	0.00	0.00	-0.25		0.00	0.00	4	0.20	0.20	0.20		0.60	0.20
5	1.00	1.00	0.00	0.00		1.00	5	0.40	0.20	0.20	0.60		1.00
6	0.50	0.00	-0.25	0.00	1.00		6	0.40	0.80	0.60	0.20	1.00	

〈표 3〉에서와 같이 설비간의 자재흐름이 가장 많은 경우와 설비간의 근접도가 가장 중요한 경우는 각각 1의 값을 가진다. 그리고 각 요인의 설비간의 차이의 비율은 그대로 유지하고 있다. 예를 들어 〈표 1〉에서 설비 2와 6의 자재흐름은 설비 1과 2의 그것 보다 2배이므로 〈표 3〉에서도 같은 비율을 가지고 있다.

다음은 각 요인의 상대적 중요도가 배치안에 반영될 수 있도록 각 요인의 가중치를 〈표 3〉의 값에다 곱한 다음 모든 요인을 하나의 혼합요인으로 만들기 위해 각 요인의 설비간의 관계값을 합하여 〈표 4〉와 같은 혼합된 관계값을

계산한다. 여기서는 질적요인의 가중치를 0.6 그리고 양적요인의 가중치를 0.4로 가정하였다.

〈표 4〉 혼합된 관계값

1	2	3	4	5	6	
1		0.61	0.39	0.08	0.76	0.46
2	0.61		0.61	0.08	0.68	0.32
3	0.39	0.61		-0.07	0.08	0.09
4	0.08	0.08	-0.07		0.24	0.08
5	0.76	0.68	0.08	0.24		1.00
6	0.46	0.32	0.09	0.08	1.00	

3. simulated annealing 알고리즘

Metropolis 등[25]은 고체가 에너지 평형상태로 변해 가는 과정을 효율적으로 시뮬레이션할 수 있는 알고리즘을 제시하였고 Kirpatrick 등[18]은 이러한 알고리즘은 조합최적화문제의 비용함수를 최소화시키는 문제에도 효율적으로 적용될 수 있음을 발견하였다.

그 후로 SA는 의사결정변수가 이산형이고 n 차원의 공간의 점으로 표시할 수 있는 순회판매원문제[2,5,19,29], 컴퓨터회로판 위에 칩을 할당하는 문제[6,7,21,33], 자재취급비용만을 고려한 설비배치문제[17,20,34] 등과 같은 여러 가지 조합최적화문제에 응용되었다.

이러한 SA의 장점은 반복적 개선 알고리즘과는 달리 목적함수를 개선할 수 없는 대안들도 특정 확률을 가지고 받아들임으로써 국부최소점을 벗어나 전체 최소점을 찾을 기회를 가질 수 있다는 점이다. SA는 annealing 절차, 상호교환을 위한 허락규칙, 멈춤규칙 그리고 매개변수 등에 의하여 영향을 받는다. 본 연구에서 사용된 각 방법들은 다음과 같다.

3. 1 annealing 절차

Annealing 절차는 초기온도(INITIME)의 설정과 시스템의 온도를 변화시키는 annealing 방법과 시점으로 구성되어 있다. 초기온도는 초기배치안과 근접한 국부최소점을 벗어날 수 있도록 적당히 큰 값을 가져야 하고 annealing 방법은 목적함수의 값을 증가시키는 배치안을 허락할 확률이 지속적으로 감소될 수 있도록 설정되어야 한다. Annealing 방법은 여러 가지

형태가 있지만 가장 단순하고 널리 사용되는 방법은 현재의 온도에 α 보다 크고 1 보다 작은 상수(FACT)를 곱하는 다음과 같은 방식이다.

$$t_i = (\text{INITIME})(\text{FACT})^i, \text{ for } i = 1, \dots, r.$$

현재의 온도, t_i 를 변화시키는 시점은 일반적으로 미리 정해진 수만큼의 상호교환을 수행한 후에 정해진 온도만큼 낮추어진다. Press 등[29]과 Jajodia 등[17]은 상호교환을 위해 시도된 수(NO_ATTEMPT)가 최대허용수(MAX_ATTEMPT)보다 작거나 또는 상호교환이 허락된 수(NO_ACCEPT)가 최대허락수(MAX_ACCEPT)보다 작으면 현재의 온도를 그대로 유지하는 방법을 사용하였다.

그러나 이러한 방법은 온도가 낮을수록 허락된 상호교환의 수는 상대적으로 작아지기 때문에 NO_ATTEMPT의 수가 MAX_ATTEMPT의 수에 도달하여도 NO_ACCEPT가 MAX_ACCEPT를 초과하지 못한다. 이런 경우에는 NO_ATTEMPT의 수가 크게 늘어날 수가 있는 단점을 가지고 있다. 따라서 본 연구에서는 NO_ATTEMPT는 MAX_ATTEMPT에서 1 보다 큰 상수(AT)를 곱한 수보다 크지 못하도록 하였다. 비록 더 많은 대안을 탐색한다는 것이 반드시 더 나은 해를 찾는다는 것은 아니지만 낮은 온도에서는 AT가 클수록 더 많은 대안을 탐색할 기회를 가지게 된다. 본 연구에서는 다음의 조건이 만족되지 않으면 현재의 온도는 평형상태에 도달하여 $i=i+1$ 로 조정되고 현재의 온도는 위의 식에 의해 낮추어진다.

$$(\text{NO_ATTEMPT} \leq \text{MAX_ATTEMPT} \text{ or } \text{NO_ACCEPT} \leq \text{MAX_ACCEPT}) \text{ and } (\text{NO_ATTEMPT} \leq \text{MAX_ATTEMPT} * \text{AT})$$

적절한 초기온도와 annealing 방법은 적용하고자 하는 문제의 목적함수의 지형(terrain)에 따라 영향을 받는다[11]. 이러한 목적함수의 지형은 사전에 알 수 없기 때문에 본 연구에서는 annealing 절차의 총 반복횟수(i)가 최소한의 수(MIN_STEP)보다 적으면 현재의 초기온도가 너무 낮은 것으로 판단하여 현재의 초기온도에다 특정 상수(INIT)를 곱하여 초기온도를 다시 설정하는 방법을 사용하였다.

3. 2 상호교환 허락기준

본 연구에서의 상호교환의 기준은 대부분의 연구에서와 같이 만약 무작위로 선택된 두개 설비의 상호교환결과가 목적함수의 감소를 초래하는 경우에는 상호교환을 허락한다. 그리고 무작위로 선택된 두개 설비의 상호교환결과가 목적함수의 증가를 초래하는 경우에는 $P(\Delta C) \cong \exp(-\Delta C/t_i)$ 의 확률을 가지고 허락한다. 여기서 ΔC 는 목적함수의 변화량 그리고 t_i 는 현재의 온도를 나타낸다. 즉, 0과 1 사이의 일양분포에서 추출된 무작위 수보다 $\exp(-\Delta C/t_i)$ 의 값이 크면 상호교환을 허락한다.

3. 3 멈춤규칙

SA 절차를 끝내는 방법은 학자들에 따라 다양하게 제시되었다. 일반적으로 사용하는 멈춤규칙은 annealing 절차의 반복횟수를 지정하는 것이다. 그러나 반복횟수를 지정하는 것 보다는 더 나은 해를 찾을 가능성이 있는 경우에는 더 많은 탐색을 그렇지 않는 경우에는 빨리 종료할 수 있도록 규칙에 유연성을 부여하는 것이 바람직하다.

Kouvelis 등[20]은 멈춤규칙으로 최소값의

변화가 없는 경우가 연속하여 특정 수 만큼 반복하는 경우 또는 특정 온도에서의 NO_ACCEPT가 특정 수 보다 작은 경우를 사용하였고, Press 등[29]과 Jajodia 등[17]은 특정 온도에서 NO_ACCEPT=0 또는 annealing 절차의 반복횟수가 특정 수에 도달한 경우를 사용하였다.

멈춤규칙은 가능성이 있는 대안을 충분히 탐색할 수 있도록 설정되어야 한다. 그러나 Kouvelis 등[20]의 규칙은 문제의 목적함수의 지형이 완만한 경우에는 대안을 충분히 탐색하지 못하고 높은 온도에서 멈춤 가능성이 있다. 따라서 본 연구에서는 3. 1절에서 설명한 비교적 낮은 온도에 발생하는 NO_ATTEMPT의 수가 $MAX_ATTEMPT*AT$ 에 도달하는 조건을 추가 하여 $NO_ATTEMPT \leq MAX_ATTEMPT*AT$ 이고 NO_ACCEPT가 최소허락수(MIN_ACCEPT) 보다 작고 최소값의 변화가 없는 경우(STOP1)가 연속하여 특정 수 (MAX_STOP1)만큼 반복하는 경우와 $NO_ATTEMPT \leq MAX_ATTEMPT*AT$ 이고 최소값의 변화가 없는 경우(STOP2)가 연속하여 특정 수(MAX_STOP2)만큼 반복하는 경우를 멈춤규칙으로 사용하였다.

이는 annealing 절차를 반복하여도 계속적으로 NO_ACCEPT의 수가 적거나 또는 최소값의 변화가 없는 경우에는 해를 더 이상 개선할 수 있는 가능성이 희박한 t_i 에 도달하였다고 판단할 수 있다. 본 연구에서의 멈춤규칙은 특정 온도 t_i 에서 평형상태에 도달한 후 앞의 두가지 조건과 Press 등[29]과 Jajodia 등[17]이 사용한 조건 중의 하나인 특정 온도에서 NO_ACCEPT=0을 추가하여 세가지 조건 중에서 하나라도 만족하면 해를 더 이상 개선할 수 있는 가능성이 희박한 t_i 에 도달하였다고 판단하여

모든 과정을 마치게 된다.

3. 4 SA 절차

본 연구에서 사용되는 SA 절차는 다음과 같고 이에 따른 흐름도는 [그림 1]에 표시되어 있다. 그리고 PASCAL 언어로 작성된 프로그램의 일부는 부록에 제시되어 있다.

단계 1: 무작위로 설비를 지역에 배치하여 최초의 현행배치안($f(C)$)을 작성하고 이에 따른 배치비용을 계산하여 현행비용(C)으로 할당한다. 최적배치안($f(C^*)=f(C)$) 그리고 최소비용($C^*=C$)로 지정한 다음 단계 2로 간다.

단계 2: 무작위로 두개의 설비를 선택한 다음 상호교환하여 임시배치안($f(C')$)을 작성하고 이에 따른 배치비용(C')을 계산한다. 시도된 배치안의 수($NO_ATTEMPT$)를 하나 증가시킨 후 다음 단계로 간다.

단계 3: 배치비용의 변화량($\Delta C=C'-C$)을 계산한다. 만약 $\Delta C < 0$ 이면 단계 5로 가고 그리고 $\Delta C \geq 0$ 이면 다음 단계로 간다.

단계 4: 확률변수 $x \sim U(0,1)$ 를 선택한다. 만약 $x < P(\Delta C) \cong \exp(-\Delta C/t_i)$ 이면 단계 5로 가고 아니면 단계 2로 간다.

단계 5: 임시배치안을 현행배치안($f(C)=f(C')$)으로 임시배치안의 비용을 현행비용($C=C'$)으로 지정한다. 그리고 상호교환이 허락된 배치안의 수(NO_ACCEPT)를 하나 증가시킨 후 단계 6으로 간다.

단계 6: 만약 $C^* < C$ 이면 $C^*=C$ 그리고 $f(C^*)=f(C)$ 로 지정하고 최소값의 변화를

알기 위해 $CHANGE=1$ 로 조정된 후 다음 단계로 간다.

단계 7: 현재의 온도 t_i 에서 평형상태에 도달하지 못하였다면 단계 2로 가고, 아니면 $i=i+1$ 그리고 $t_i=t_i \times FACT$ 로 조정된 후 다음 단계로 간다.

단계 8: $NO_ATTEMPT \leq MAX_ATTEMPT \times AT$ 이고 $CHANGE=0$ 이고 $NO_ACCEPT \leq MIN_ACCEPT$ 이면 $STOP1=STOP1+1$ 으로 조정한다. 그리고 $NO_ATTEMPT \leq MAX_ATTEMPT \times AT$ 이고 $CHANGE=0$ 이면 $STOP2=STOP2+1$ 로 조정된 후 다음 단계로 간다.

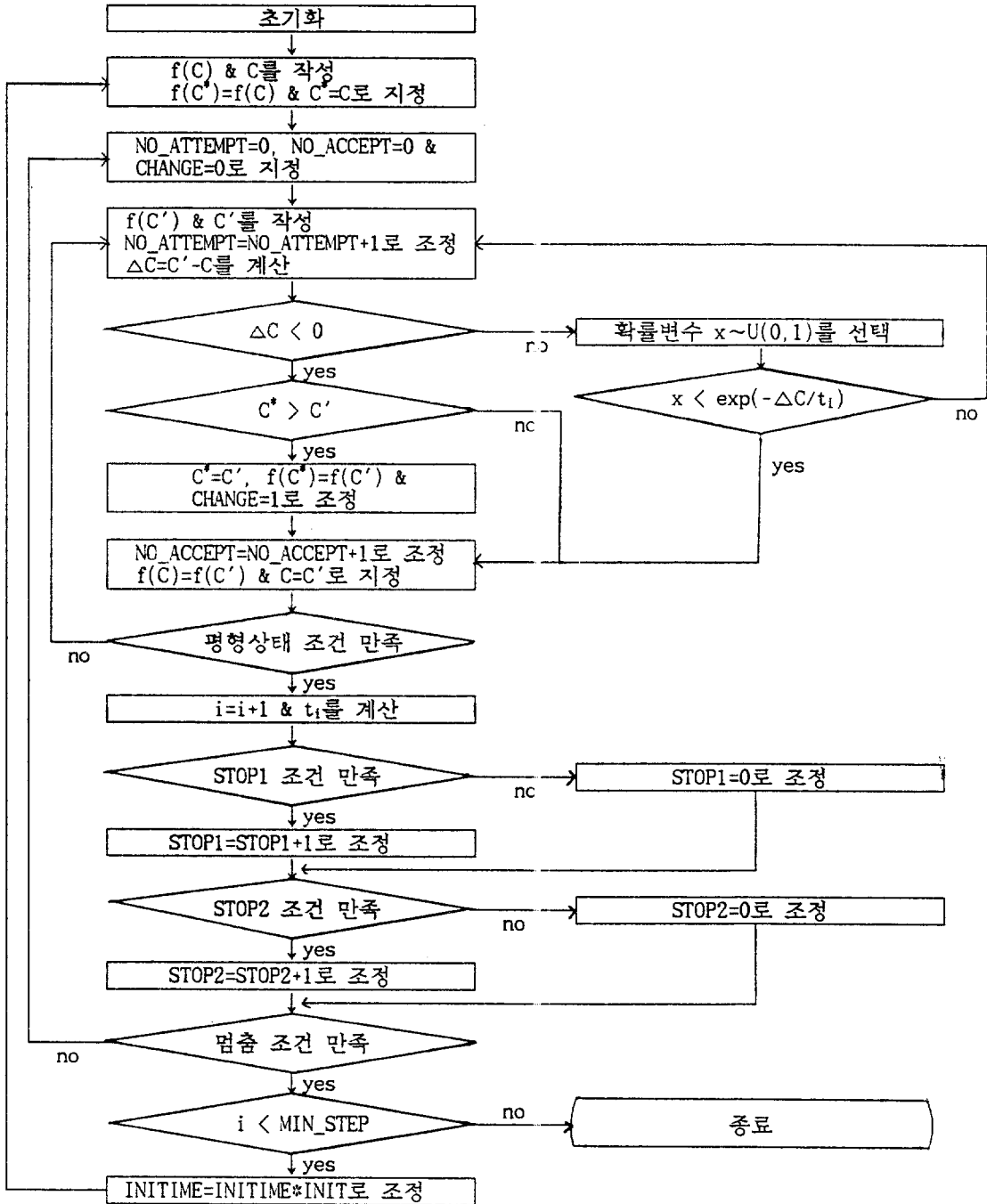
단계 9: 이 단계에서 $NO_ACCEPT=0$ 또는 $STOP1=MAX_STOP1$ 또는 $STOP2=MAX_STOP2$ 이면 단계 10으로 가고, 아니면 $NO_ATTEMPT=0$, $NO_ACCEPT=0$ 그리고 $CHANGE=0$ 으로 조정된 후 단계 2로 간다.

단계 10: 만약 총 반복횟수 i 가 MIN_STEP 보다 크면 멈추고 작으면 초기온도를 $INITIME \times INIT$ 로 조정하여 단계 1로 되돌아 간다.

4. 실험결과와 비교분석

본 연구에서 제안된 모형과 SA 알고리즘의 효율성을 측정하기 위하여 자재흐름과 근접도를 동시에 고려한 3개의 다수요인문제[10,13,32] 그리고 자재흐름만을 고려한 8개의 문제[27,34]를 사용하였다. 이러한 문제들에 적용하

[그림 1] SA 절차의 흐름도



기 위해 SA 절차에 사용된 매개변수들의 값들은 INITIME=10, FACT=0.9, INIT=5, MIN_STEP=3, AT=3, MAX_STOP1=3, MAX_STOP2=10, MAX_ATTEMPT=n*10, MAX_ACCEPT=30, 그리고 MIN_ACCEPT=MAX_ACCEPT/2이다. INITIME와 FACT의 값은 기존연구[17,20,24,29]에서 사용된 값을 사용하였고 그리고 나머지 값들은 기존의 연구에서 제시된 값을 근거로 하여 여러 가지 실험을 통해 가장 적절하다고 판단되는 값을 사용하였다.

4. 1 알고리즘의 효율성에 대한 실험결과

다수요인을 고려한 모형에서는 배치비용의 계산방법이 서로 다르기 때문에 문헌에서 제시한 결과들을 직접 비교하기가 불가능하다. 따라서 다수요인에 대한 SA 알고리즘의 효율성을 측정하기 위해 Fortenber & Cox[10]의 연구(F&C), Urban[32]의 연구(U), 그리고 Harmonosky와 Tothero[13]의 연구(H&T)가 제시한 결과, 즉 각 연구자의 모형과 알고리즘을 적용한 배치비용과 각 연구자의 모형에다 본 연구에서 제안된 SA 알고리즘을 적용한 배치비용을 먼저 비교하였다. 이에 대한 결과는 <표 5>에 제시되어 있다.

<표 6>은 본 연구의 모형과 각 연구자의 알고리즘을 적용한 배치비용과 본 연구의 모형과 본 연구의 알고리즘을 적용한 배치비용을 나타낸다. <표 5>와 <표 6>에서와 같이 모든 유형의 문제에서 SA의 결과가 기존의 결과와 같거나 우수한 것으로 나타났다. 이는 본 연구에서 제시된 SA의 절차는 여러 가지 모형에서도 잘 적용될 수 있다는 것을 보여주고 있다.

Kusiak와 Heragu[22]가 제시한 다음과 같은 식을 이용하여 결과의 질(solution quality: SQ)을 각 연구자의 모형과 알고리즘을 사용하여 계산한 SQ의 결과와 본 연구에서 제안된 모형과 SA 알고리즘을 사용하여 계산한 SQ의 결과를 서로 비교하였다.

$$SQ = \frac{E}{LB} \quad (100)$$

여기서 E는 최종배치안에 따른 목적함수값을 그리고 LB는 하한값으로 $n(n-1)/2$ 개의 설비들 간의 관계값들이 내림순으로 정렬된 값들(T)과 $n(n-1)/2$ 개의 거리값들이 올림순으로 정렬된 값들(d)을 사용하여 다음의 식에 의해 계산한다. 이에 따른 비교 결과는 <표 7>에 제시되어 있다.

$$LB = \sum_{i=1}^{n-1} \sum_{j=i+1}^n T_i d_j$$

<표 7>에서 SQ2가 SQ1보다 모든 유형의 문제에서 값이 더 작으므로 본 연구에서 제시된 모형과 SA 알고리즘을 사용할 때가 기존의 모형과 알고리즘을 사용할 때보다 하한값에 더 가까운 해를 찾을 수 있는 것으로 나타났다.

Harmonosky와 Tothero[13]은 초기배치안작성과 개선해 알고리즘을 동시에 사용하였기 때문에 <표 5>에서 <표 7>까지의 H&T의 문제유형은 각 요인의 가중치를 서로 다르게 부여한 경우이고, 그리고 Fortenber & Cox[10]와 Urban[32]은 개선해 알고리즘만을 사용하였기 때문에 F&C와 U의 문제유형은 초기배치안이 서로 다른 경우이다. <표 5>와 <표 6>에서 F&C의 n=8인 경우의 비용1의 값은 초기배치안에 따라 영향을 받는 greedy 알고리즘을 사

〈표 5〉 다수요인문제들의 실험결과 1

연구자	문제 유형	n=6		n=8		n=12	
		비용1	비용2	비용1	비용2	비용1	비용2
H&T	1	1.437	1.437	1.598	1.467	1.991	1.980
	2	1.483	1.441	1.74	1.490	2.092	1.967
	3	1.445	1.445	1.63	1.513	2.116	1.934
	4	1.467	1.448	1.65	1.536	2.002	1.882
	5	1.445	1.436	1.698	1.560	2.000	1.834
	6	1.424	1.381	1.66	1.512	1.946	1.772
	7	1.402	1.326	1.62	1.459	1.892	1.700
	8	1.380	1.271	1.60	1.405	1.939	1.619
	9	1.215	1.215	1.57	1.352	1.808	1.601
	10	1.160	1.160	1.54	1.299	1.785	1.538
	11	1.316	1.105	1.50	1.245	***	***
	평균	1.380	1.333	1.62	1.440	1.957	1.783
F&C	1	368	368	1028	1028	4540	4464
	2	368	368	1058	1028	***	***
	3	368	368	1058	1028	***	***
	4	368	368	1028	1028	***	***
	평균	368	368	1048	1028	4540	4464
U	1	768	768	***	***	7448	7324
	2	986	986	***	***	***	***
	평균	877	877	***	***	7448	7324

비용1: 각 연구자의 모형과 알고리즘을 적용한 배치비용
 비용2: 각 연구자의 모형과 본 연구의 SA 절차를 적용한 배치비용
 ***: 기존 문헌에서 제시하지 않았기 때문에 비교대상에서 제외한 경우

〈표 6〉 다수요인문제들의 실험결과 2

연구자	문제 유형	n=6		n=8		n=12	
		비용1	비용2	비용1	비용2	비용1	비용2
H&T	1	18.400	18.400	39.000	35.800	117.455	115.091
	2	17.960	17.960	42.120	36.720	107.573	104.800
	3	17.520	17.520	42.140	37.640	97.836	93.600
	4	17.080	17.080	42.760	38.560	88.864	87.954
	5	16.640	16.640	43.040	39.240	81.764	81.264
	6	15.850	15.850	42.700	38.200	75.573	73.823
	7	14.780	14.780	42.360	37.160	68.418	66.745
	8	13.710	13.710	42.020	36.120	60.959	58.618
	9	12.640	12.640	41.380	35.080	56.745	53.749
	10	11.570	11.570	34.040	34.040	56.500	50.000
	11	12.500	10.500	33.000	33.000	***	***
	평균	15.332	15.150	40.169	36.505	81.169	78.564
F&C	1	32.100	31.700	82.200	76.400	177.090	176.000
	2	32.100	31.700	80.300	76.400	***	***
	3	31.700	31.700	80.300	76.400	***	***
	4	32.100	31.700	82.100	76.400	***	***
	평균	32.000	31.700	81.550	76.400	177.090	176.000
U	1	38.722	37.944	***	***	180.409	176.000
	2	28.167	28.167	***	***	***	***
	평균	33.445	33.056	***	***	180.409	176.000

비용1: 본 연구의 모형과 각 연구자의 알고리즘을 적용한 배치비용
 비용2: 본 연구의 모형과 본 연구의 SA 절차를 적용한 배치비용
 ***: 기존 문헌에서 제시하지 않았기 때문에 비교대상에서 제외한 경우

〈표 7〉 다수요인문제들의 실험결과 3

연구자	문제 유형	n=6		n=8		n=12	
		SQ1	SQ2	SQ1	SQ2	SQ1	SQ2
H&T	1	104.545	104.545	100.000	100.000	116.058	115.511
	2	104.859	104.784	109.117	100.685	113.034	111.586
	3	106.499	105.542	109.788	101.950	118.166	110.047
	4	110.365	107.692	111.529	104.244	115.969	111.175
	5	114.357	111.828	111.548	106.630	115.579	112.866
	6	116.355	114.855	105.521	105.379	118.813	114.818
	7	119.301	116.930	104.819	104.676	123.085	119.111
	8	123.265	120.580	104.074	103.942	128.015	124.012
	9	127.899	125.397	103.568	103.542	131.151	127.023
	10	133.390	131.627	103.355	103.339	133.766	129.870
	11	140.000	140.000	103.125	103.125	***	***
평균	118.258	116.707	106.040	103.410	121.364	117.602	
F&C	1	117.949	114.855	107.307	105.379	117.192	111.577
	2	117.949	114.855	109.812	105.379	***	***
	3	117.949	114.855	109.812	105.379	***	***
	4	117.949	114.855	107.307	105.379	***	***
	평균	117.949	114.855	108.560	105.379	117.192	111.577
U	1	111.628	109.937	***	***	120.831	114.565
	2	110.787	105.625	***	***	***	***
	평균	111.207	107.781	***	***	120.831	114.565

SQ 1: 각 연구자의 모형과 알고리즘을 적용하여 QS를 계산한 결과

SQ 2: 본 연구의 모형과 SA 절차를 적용하여 QS를 계산한 결과

***: 기존 문헌에서 제시하지 않았기 때문에 비교대상에서 제외한 경우

용하였기 때문에 문제유형 2와 3은 주어진 초기배치안과 근접한 국부적 최소해에 벗어나지 못한 경우이고 문제유형 1과 4는 전체 최소점을 찾은 경우이다. 〈표 7〉의 F&C의 n=8인 경우의 SQ1의 값은 〈표 5〉의 비용 1의 값에 의하여 계산한 결과이기 때문에 같은 형태를 가진다.

〈표 8〉은 Wilhelm와 War[34]가 그들이 개발한 SA 알고리즘의 효율성을 측정하기 위하여 Nugent 등[27]의 자재취급비용만을 최소화하는 단일요인을 가진 8개의 문제를 기존의 알고리즘의 결과와 비교해 놓은 표에 같은 조건 하에서 본 연구의 실험결과와 각 결과의 범위를 계산하여 첨부한 표이다.

설비의 수가 12개인 문제를 제외하고는 본 연구의 평균이 다른 알고리즘의 평균과 같거나 우수하다. 또한 초기배치안에 따른 결과의 변동범위를 보면 설비의 수가 20개인 문제를 제외하고는 본 연구에서 제시한 결과의 범위가 가장 작으므로 본 연구에서 제안한 SA 알고리즘은 초기배치에 크게 영향을 받지 않는 것으로 나타났다.

CPU 시간은 프로그래밍 기술, 사용한 컴퓨터의 종류 등과 같은 외부적인 요인에 의해 영향을 받기 때문에 CPU 시간의 직접적인 비교는 불가능하다. 본 연구의 CPU 시간은 개인용 컴퓨터 486 DX2-50에서 PASCAL 언어로 코딩하여 측정된 값이다. CPU 시간의 변동폭은

〈표 8〉 단일요인문제들의 실험결과

설비 수	초기 배치안 번호	Wilhelm & Ward		CRAFT		CRAFT Sampling		Revised Hiller		본 연구	
		결과	CPU Time	결과	CPU Time	결과	CPU Time	결과	CPU Time	결과	CPU Time
5	1	25.0	1.87	29.0	0.56	29.0	1.99	25.0		25.0	0.39
	2	25.0	1.87	29.0	0.55	29.0	1.87	25.0		25.0	0.39
	3	25.0	1.87	25.0	0.56	25.0	1.98	25.0		25.0	0.39
	4	25.0	1.87	29.0	0.58	29.0	2.08	25.0		25.0	0.38
	5	25.0	1.87	29.0	0.59	26.0	2.06	25.0		25.0	0.39
	평균 범위	25.0	1.87	28.2	0.57	27.6	2.00	25.0	0.24	25.0	0.39
		0.0	0.00	4.0	0.04	4.0	0.21	0.0		0.0	0.01
6	1	43.0	2.95	43.0	0.54	43.0	1.92	43.0		43.0	0.39
	2	43.0	2.94	43.0	0.59	43.0	1.92	43.0		43.0	0.38
	3	43.0	2.74	46.0	0.56	46.0	1.38	43.0		43.0	0.39
	4	43.0	2.74	43.0	0.57	43.0	1.90	43.0		43.0	0.39
	5	43.0	2.92	46.0	0.61	43.0	2.02	43.0		43.0	0.38
	평균 범위	43.0	2.86	44.2	0.57	43.6	1.91	43.0	0.24	43.0	0.39
		0.0	0.21	3.0	0.07	3.0	0.64	0.0		0.0	0.01
7	1	74.0	4.13	79.0	0.73	76.0	2.63	72.0		74.0	0.55
	2	74.0	3.45	78.0	0.81	74.0	3.19	72.0		74.0	0.61
	3	74.0	4.27	74.0	0.78	74.0	3.11	76.0		74.0	0.60
	4	74.0	3.83	84.0	0.79	74.0	3.07	74.0		74.0	0.55
	5	74.0	3.37	83.0	0.71	76.0	2.75	72.0		74.0	0.66
	평균 범위	74.0	3.81	79.6	0.76	74.8	2.95	73.2	0.55	74.0	0.60
		0.0	0.90	10.0	0.10	2.0	0.56	4.0		0.0	0.11
8	1	107.0	4.99	119.0	0.72	113.0	2.98	107.0		107.0	0.86
	2	107.0	5.05	107.0	0.81	117.0	3.29	107.0		107.0	1.02
	3	107.0	3.84	107.0	0.72	117.0	2.82	109.0		107.0	1.08
	4	107.0	4.99	110.0	0.71	117.0	2.98	107.0		107.0	1.02
	5	107.0	5.07	107.0	0.73	117.0	2.98	107.0		107.0	1.08
	평균 범위	107.0	4.79	110.0	0.74	118.2	3.01	107.4	0.44	107.0	1.01
		0.0	1.23	12.0	0.10	6.0	0.47	2.0		0.0	0.22
12	1	289.0	6.59	298.0	1.32	276.0	7.06	304.0		293.0	1.85
	2	291.0	7.32	308.0	1.33	278.0	6.82	293.0		289.0	1.74
	3	293.0	6.65	291.0	1.46	276.0	7.72	307.0		293.0	2.40
	4	293.0	7.56	295.0	1.52	275.0	8.28	297.0		293.0	2.34
	5	289.0	6.54	289.0	1.33	279.0	5.32	291.0		289.0	2.24
	평균 범위	291.0	6.93	296.2	1.39	274.8	7.04	298.4	1.35	291.4	2.11
		4.0	1.02	19.0	0.20	9.0	2.96	16.0		4.0	0.66
15	1	576.0	15.53	628.0	1.83	578.0	11.36	596.0		576.0	3.83
	2	575.0	9.40	588.0	2.44	570.0	17.10	580.0		576.0	4.21
	3	576.0	11.49	591.0	2.41	570.0	15.38	575.0		580.0	4.21
	4	584.0	10.72	640.0	2.01	575.0	14.87	576.0		575.0	3.33
	5	580.0	11.02	583.0	2.39	575.0	18.31	584.0		580.0	4.10
	평균 범위	578.2	11.63	606.0	2.22	571.0	15.40	582.2	2.38	577.4	3.94
		9.0	6.13	57.0	0.61	13.0	6.95	21.0		5.0	0.88
20	1	1303.0	18.67	1334.0	5.81	1377.0	41.00	1328.0		1287.0	9.32
	2	1313.0	15.45	1354.0	4.05	1377.0	35.50	1297.0		1294.0	10.52
	3	1315.0	20.75	1351.0	4.71	1378.0	34.60	1348.0		1300.0	13.49
	4	1300.0	20.14	1324.0	4.57	1371.0	39.60	1328.0		1311.0	10.69
	5	1309.0	19.10	1332.0	4.91	1372.0	41.80	1322.0		1300.0	11.74
	평균 범위	1308.0	18.82	1339.0	4.81	1371.0	39.50	1324.6	5.83	1298.4	11.15
		15.0	5.30	30.0	1.76	9.0	7.20	51.0		24.0	4.17
30	1	3064.0	30.85	3090.0	28.54	3100.0	220.00	3070.0		3109.0	29.11
	2	3111.0	43.75	3192.0	19.67	3170.0	217.00	3147.0		3095.0	31.31
	3	3095.0	49.24	3197.0	20.48	3160.0	204.00	3153.0		3101.0	27.57
	4	3079.0	40.15	3273.0	15.68	3190.0	202.00	3124.0		3094.0	26.26
	5	3150.0	47.35	3237.0	18.86	3160.0	214.00	3077.0		3093.0	29.11
	평균 범위	3099.8	42.27	3197.8	20.65	3140.0	211.00	3114.2	22.86	3098.4	28.67
		86.0	18.39	183.0	12.68	60.0	18.00	83.0		16.0	5.05

전반적으로 CRAFT 알고리즘이 제일 적고 다음이 본 연구의 변동폭이다. 설비의 수가 증가함에 따른 본 연구의 CPU 시간의 증가율은 CRAFT와 Revised Hiller 알고리즘의 증가율과 비슷한 형태를 보이고 있다.

Kouvelis 등[20]은 그들이 제안한 SA는 무작위로 선택된 초기배치안을 사용한 결과보다는 초기배치안작성(construction) 알고리즘을 사용하여 작성한 초기배치안을 사용한 경우가 더 우수하다고 하였다. 그러나 본 연구에서의 이와 같은 실험결과는 초기배치안작성 알고리즘을 사용한 경우가 평균적으로 보아 우수한 해를 제시하지는 못하였다. 이는 앞의 결과와 같이 본 연구의 SA 알고리즘은 초기해에 큰 영향을 받지 않는 것으로 나타났다.

<표 5>, <표 6> 그리고 <표 8>에 있는 128개 유형의 문제 중에서 <표 5>에서 Fortenber & Cox[10]의 n=8인 4개의 문제와 n=12인 경우, 그리고 Urban[32]의 n=12인 문제등 6개의 문

제에서 annealing 절차의 반복횟수가 3보다 작아 초기온도가 10에서 50으로 조정되어 새로운 annealing 절차가 한번 반복되었다. 초기온도가 조정되기 전의 배치비용은 각각 1052, 1052, 1052, 1028, 4620, 그리고 7448이었다. 이러한 문제들의 ΔC 의 값들이 다른 나머지 문제들의 ΔC 의 값들보다 크기 때문에 처음의 초기온도가 너무 낮아 상호교환을 허락받을 확률이 작아서 annealing 절차의 반복횟수가 3이 되기 전에 NO_ACCEPT가 0이 되었다. 이들 문제들의 시도된 배치안의 ΔC 의 평균은 250이상 이었고 나머지 문제유형의 평균은 0.3에서 80 사이였다.

4. 2 모형의 효율성에 대한 실험결과

본 연구에서 제안된 모형의 효율성을 측정하기 위하여 Fortenber & Cox[10]의 모형 (F&C), Urban[32]의 모형(U), Harmonosky

<표 9> 본 연구의 SA 알고리즘을 사용한 모형의 실험결과

설비 수	α 범위	자재취급비용				근접비용			
		F&C	U	H&T	H	F&C	U	H&T	H
6	0.0-0.2	212.0	189.6	184.0	184.0	114.0	53.2	56.0	56.0
	0.2-0.4	212.0	212.0	189.6	184.0	112.0	42.0	53.2	56.0
	0.6-0.8	212.0	212.0	212.0	212.0	112.8	42.0	42.0	42.0
	0.8-1.0	212.0	212.0	212.0	212.0	109.6	42.0	42.0	42.0
8	0.0-0.2	434.0	381.2	358.0	358.0	284.0	166.8	180.0	180.0
	0.2-0.4	418.8	426.8	358.0	373.2	288.8	138.0	180.0	170.4
	0.6-0.8	434.0	426.8	434.0	434.0	284.0	138.0	132.0	132.0
	0.8-1.0	434.0	426.8	434.0	434.0	282.4	138.0	132.0	132.0
12	0.0-0.2	2553.6	2602.4	2544.0	2543.6	585.6	260.4	262.4	266.0
	0.2-0.4	2589.6	2784.8	2575.6	2552.0	566.8	217.6	236.0	240.0
	0.6-0.8	2588.8	2925.2	2825.6	2790.8	578.0	212.0	204.8	206.0
	0.8-1.0	2588.0	2854.8	2952.4	2955.2	600.4	208.8	201.6	201.2

<표 10> CRAFT 알고리즘을 사용한 모형의 실험결과

설비 수	α 범위	자재취급비용				근접비용			
		F&C	U	H&T	H	F&C	U	H&T	H
6	0.0-0.2	212.0	197.6	193.6	192.0	117.6	50.0	52.0	52.0
	0.2-0.4	212.0	212.0	196.0	195.2	113.6	42.0	50.0	52.0
	0.6-0.8	212.0	212.0	212.0	212.0	115.2	42.0	42.0	42.0
	0.8-1.0	212.0	212.0	212.0	212.0	114.8	42.0	42.0	42.0
8	0.0-0.2	406.0	396.4	367.6	367.6	292.0	173.6	178.4	178.4
	0.2-0.4	406.0	406.0	406.0	406.0	292.0	172.0	172.0	172.0
	0.6-0.8	406.0	406.0	406.0	406.0	292.0	172.0	172.0	172.0
	0.8-1.0	406.0	406.0	406.0	406.0	292.0	172.0	172.0	172.0
12	0.0-0.2	2726.0	2694.4	2645.6	2630.0	564.0	234.8	251.6	258.0
	0.2-0.4	2726.0	2726.0	2724.0	2689.6	564.0	226.0	227.6	239.2
	0.6-0.8	2726.0	2770.0	2761.2	2734.8	564.0	222.0	222.8	225.2
	0.8-1.0	2726.0	2816.4	2786.4	2816.4	564.0	221.6	223.6	221.6

와 Tothero[13]의 모형(H&T), 그리고 본 연구에서 제시된 모형(H)을 본 연구의 SA 알고리즘과 CRAFT 알고리즘을 사용하여 작성된 최종배치안에 따른 자재취급 비용과 근접비용을 계산하여 각 요인들의 상대적 중요도를 나타내는 가중치 α 를 반영하는 정도를 간접적인 방법으로 서로 비교하였다.

최종배치안을 가지고 자재취급비용은 설비들간의 자재흐름량에다 거리를 곱하여 계산하였고 그리고 근접비용은 설비들간의 근접점수에다 거리를 곱하여 계산하였다. 이와 같이 본 연구의 SA 알고리즘을 가지고 계산한 결과는 <표 9>에 그리고 CRAFT 알고리즘을 사용한 결과는 <표 10>에 제시되어 있다.

<표 9>와 <표 10>에서 α 의 범위는 근접도요인에 부여한 가중치이므로 자재흐름요인에 대한 가중치는 $1-\alpha$ 이고, 표의 값들은 주어진 α 의 범위내에서 여러 가지 α 값을 사용하여 작성된 배치안에 따른 자재취급비용과 근접비용을 제

산하여 평균한 값이다.

α 가 0.5 이상이면 자재흐름보다 근접도요인에 더 높은 비중을 부여하였기 때문에 α 가 0.5 이하일 때 보다 근접비용은 적어져야 하고 자재취급비용은 커져야 한다. 반대로 α 가 0.5 이하이면 근접도보다 자재취급요인에 더 높은 비중을 두기 때문에 α 가 0.5 이상일 때 보다 자재취급비용은 작아져야 하고 근접비용은 커져야 한다. 따라서 α 가 증가함에 따라 자재취급비용은 증가하여야 하고 반대로 근접비용은 감소하여야 한다.

<표 9>에서 이와 같은 관계를 세가지 문제에 대하여 F&C 모형은 0개, U 모형은 2개, 그리고 H&T와 H 모형은 3가지 모두에 대하여 이러한 관계를 잘 반영하고 있다. <표 10>에서는 F&C 모형은 0개, U 모형은 3개, H&T 모형은 2개, 그리고 H 모형은 3개이다. <표 9>와 <표 10>에서 F&C 모형은 α 에 관계없이 자재취급비용과 근접비용이 큰 차이를 보이지 않고

있다. F&C 모형은 나머지 세 모형과 다른 근접점수를 사용하였으나 세 모형과 같은 근접점수를 사용하여도 비슷한 결과를 보여 주었다. U 모형은 <표 9>에서 설비가 12개인 경우에 자재취급비용이 증가하다가 감소하였고, 그리고 H&T 모형은 <표 10>에서 설비가 12개인 경우에 근접비용이 감소하다가 증가하였다.

α 가 0.2 이하인 경우에는 자재취급비용이 적을수록 그리고 α 가 0.8 이상인 경우에는 근접비용이 적을수록 바람직하다. F&C 모형을 제외하고, <표 9>에서 α 가 0.2 이하일 때 자재취급비용이 최소인 경우와 α 가 0.8 이상일 때 근접비용이 최소인 경우의 합은 U 모형이 1번, H&T 모형이 4번, 그리고 H 모형은 모든 경우에 최소인 6번이다. 가중치가 0.2~0.4 그리고 0.6~0.8의 경우에는 U 모형이 1번, H&T와 H 모형은 4번이다. <표 10>에서 모든 경우의 수의 합은 U 모형이 6번, H&T 모형은 5번, 그리고 H 모형은 11번이다.

따라서 본 연구의 모형이 각 요인에 대한 α 의 값을 가장 잘 반영하고 있다고 할 수 있다. 배치비용을 계산할 때 자재흐름과 근접점수를 F&C 모형은 곱하는 형태를, U 모형은 각 요인의 관계값을 표준화하지 않고 더하는 형태를, H&T 모형은 각 요인의 관계값의 합이 1이 되도록 표준화하여 더하는 형태를 취하고 있다. 따라서 각 요인이 배치안에 동등한 영향을 미칠 수 있도록 원래의 값을 표준화한 다음에 각 요인의 가중치를 반영하는 것이 바람직하다.

5. 결론

필요한 설비나 기계를 적정장소에 배치하는 문제는 공장에서 발생하는 중요한 의사결정문제이다. 본 연구에서는 이러한 배치문제를 해결할 수 있는 다수요인을 고려한 모형과 SA 알고리즘을 개발하여 그 효율성을 측정하기 위해 기존의 연구들과 비교하였다. 그 결과 다수요인을 고려한 모형은 각 요인의 중요도를 다른 모형에 비하여 더 잘 반영하였고 그리고 SA 알고리즘은 다른 알고리즘에 비해 초기배치안에 크게 영향을 받지 않고도 더 우수한 해를 제시할 수 있었다.

그러나 본 연구에서는 자재흐름과 근접도의 두 가지 요인만을 고려하였으나 두 가지 이상의 요인과 30개 이상의 설비를 가진 문제들에 대한 실험과 분석이 필요하다. 또한 본 연구에서는 설비의 크기나 모양이 일정한 것(equidimensional facilities)으로 가정하였으나 이는 현실과는 거리가 있다. 따라서 크기나 모양을 고려한 모형과 알고리즘에 대한 연구가 필요하다.

참고 문헌

- [1] Armour, G. C. and E. S. Buffa, "A Heuristic Algorithm and Simulation Approach to Relative Location of Facilities," *Management Science*, Vol. 9, No. 2 (1963), pp. 294-309.
- [2] Bonomi, E. and J. Lutton, "The N-City Travelling Salesman Problem: Statistical Mechanics and the Metropolis Algorithm,"

- ithm," *SIAM Review*, Vol. 26, No. 4 (1984), pp. 551-568.
- [3] Buffa, E. S., G. C. Armour and T. E. Vollman, "Allocating Facilities with CRAFT," *Harvard Business Review*, Vol. 42(1964), pp. 136-159.
- [4] Burkard, R. E., "Locations with Spatial Interaction-Quadratic Assignment Problem," in R. L. Francis and P. B. Mirchandani (Eds.), *Discrete Location Theory*, Prentice Hall, Englewood Cliffs, N. J., 1984.
- [5] Burkard, R. E. and F. Rendl, "A Thermodynamically Motivated Simulation Procedure for Combinatorial Optimization Problems," *European Journal of Operational Research*, Vol. 17(1984), pp. 169-174.
- [6] Casotto, A., F. Romeo and A. Sangiovanni-Vincentelli, "A Parallel Simulated Annealing Algorithm for the Placement of Macro-Cells," *IEEE Transactions on Computer-Aided Design*, CAD-6, 5(1987), pp. 838-847.
- [7] Darema, F., S. Kirkpatrick and V. A. Norton, "Parallel Algorithm for Chip Placement by Simulated Annealing," *IBM Journal of Research and Development*, Vol. 31, No. 3(1987), pp. 391-402.
- [8] Dutta, K. N. and S. Sahu, "A Multi-goal Heuristic for Facilities Design Problems: MUGHAL," *International Journal of Production Research*, Vol. 20, No. 2(1982), pp. 147-154.
- [9] Edwards, H. K., B. E. Gillett and M. E. Hale, "Modular Allocation Technique(MAT)," *Management Science*, Vol. 17, No. 3(1970), pp. 161-169
- [10] Fortenberr, J. C. and J. F. Cox, "Multiple Criteria Approach to the Facilities Layout Problem," *International Journal of Production Research*, Vol. 23, No. 4(1985), pp. 773-782.
- [11] Haddock, J. and J. Mittenthal, "Simulation Optimization Using Simulated Annealing," *Computers & Industrial Engineering*, Vol. 22, No. 4(1992), pp. 387-395.
- [12] Ham, I., K. Hitomi and T. Yoshida, *Group Technology: Applications to Production Management*, Kluwer-Nijhoff Publishing, Boston, 1985.
- [13] Harmonosky, C. M. and G. K. Tothoro, "A Multi-factor Plant Layout Methodology," *International Journal of Production Research*, Vol. 30, No. 8(1992), pp. 1773-1789.
- [14] Hassan, M. M. D., G. L. Hogg and D. R. Smith, "SHAPE: A Construction Algorithm for Area Placement Evaluation," *International Journal of Production Research*, Vol. 24, No. 2(1986), pp. 1283-1295.
- [15] Hiller, F. S., "Quantitative Tools for Plant Layout Analysis," *Journal of Industrial Engineering*, Vol. 14(1963), pp. 33-40.
- [16] Hiller, F. S. and M. M. Connors, "Quadratic Assignment Problem Algorithms and the Location of Indivisible Facili-

- ties," *Management Science*, Vol. 13, No. 1(1966), pp. 42-57.
- [17] Jajodia, S., I. Minis, G. Harhalakis and J. Proth, "CLASS: Computerized Layout Solutions Using Simulated Annealing," *International Journal of Production Research*, Vol. 30, No. 1(1992), pp. 95-108.
- [18] Kirkpatrick, S., C. D. Gelatt, Jr. and M. P. Vecchi, "Optimization by Simulated Annealing," *Science*, Vol. 220 (1983), pp. 671-680.
- [19] Kirkpatrick, S., "Optimization by Simulated Annealing: Quantitative Studies," *Journal of Statistical Physics*, Vol. 34(1984), pp. 975-986.
- [20] Kouvelis, P. and W. Chiang, "A Simulated Annealing Procedure for Single Row Layout Problems in Flexible Manufacturing Systems," *International Journal of Production Research*, Vol. 30, No. 4(1992), pp. 717-732.
- [21] Kravitz, S. A. and R. A. Rutenbar, "Placement by Simulated Annealing on a Microprocessor," *IEEE Transactions on Computer-Aided Design*, CAD-6, 4(1987), pp. 534-549.
- [22] Kusiak, A. and S. S. Heragu, "The Facility Layout Problem," *European Journal of Operational Research*, Vol. 29 (1987), pp. 229-251.
- [23] Lee, R. C. and J. M. Moore, "CORELAP-Computerized Relationship Layout Planning," *Journal of Industrial Engineering*, Vol. 18, No. 3(1967), pp. 195-200.
- [24] Levary, R. R., "Facilities Layout-A Survey of Solution Procedures," *Computer & Industrial Engineering*, Vol. 9, No. 2 (1985), pp. 141-145.
- [25] Metroplolis, N., A. N. Rosenbluth, M. N. Rosenbluth, A. H. Teller and E. Teller, "Equation of State Calculation by Fast Computing Machines," *Journal of Chemical Physics*, Vol. 21, No. 6 (1953), pp. 1087-1092.
- [26] Muther, R., *Systematic Layout Planning*, Cahers Books, Boston, 1973.
- [27] Nugent, C. E., T. E. Vollman and J. Ruml, "An Experimental Comparison of Techniques for the Assignment of Facilities to Locations," *Operations Research*, Vol. 16(1968), pp. 150-173.
- [28] Picone, C. J. and W. E. Wilhelm, "A Perturbation Scheme to Improve Hiller's Solution to the Facilities Location Problem," *Management Science*, Vol. 30, No. 10(1984), pp. 1238-1249.
- [29] Press, W. H., B. P. Flannery, S. A. Teukolsky and W. T. Vetterling, *Numerical Recipes*, Cambridge University Press, New York, 1986.
- [30] Rosenblatt, M. J., "The Facilities Layout Problem: A Multi-goal Approach," *International Journal of Production Research*, Vol. 17, No. 4(1979), pp. 323-332.
- [31] Seehof, J. M. and W. O. Evans, "Automated Layout Design Program," *Journal of Industrial Engineering*, Vol. 18, No. 12(1967), pp. 690-695.

-
- [32] Urban, T. L., "A Multiple Criteria Model for the Facilities Layout Problem," *International Journal of Production Research*, Vol. 25, No. 12(1987), pp. 1805-1812.
- [33] Vecchi, M. P. and S. Kirkpatrick, "Global Wiring by Simulated Annealing," *IEEE Transactions on Computer-Aided Design*, CAD-2, 4(1983), pp. 215-222.
- [34] Wilhelm, M. and T. Ward, "Solving Quadratic Assignment Problems by Simulated Annealing," *IIE Transactions*, (1987), pp. 107-119.

[부록] PASCAL 프로그램의 일부

```

procedure makesucc(lay:layoutype);
var i:integer;
begin
  no_accept:=no_accept+1;
  mincost:=cost;
  for i := 1 to no_facility do layout[i]:=lay[i];
end;
procedure swap;
var i,j:integer;
  temp:layoutype;
begin
  u:=0;v:=0;
  while (u=0) or (v=0) or (u=v) do
  begin
    u:=random(no_facility+1);
    v:=random(no_facility+1);
  end;
  for i := 1 to no_facility do temp[i]:=layout[i];
  for i:= 1 to no_facility do
  begin
    if u=temp[i] then temp[i]:=v
    else if v=temp[i] then temp[i]:=u;
  end;
  makecost(temp);
  if cost < minimum then
  begin
    makesucc(temp);
    for i:= 1 to no_facility do finalay[i]:=lay[i];
    minimum:=mincost;
    change:=true;
  end
  else if cost < mincost then makesucc(temp)
  else if random < exp(-(cost-mincost)/ttt) then makesucc(temp);
  no_attempt:=no_attempt+1;
end;
procedure control;
label 999;
begin
  clockon;
  initime:=10; fact:=0.9;
  init:=5; min_step:=3;
  at:=3; max_stop1:=3;
  max_stop2:=10; max_attempt:=no_facility*10;
  max_accept:=30; min_accept:=max_accept div 2;

```

```
999: makeinit:
  ttt:=initime;
  no_temp_step:=1;
  stop1:=0;
  stop2:=0;
  repeat
    no_attempt:=0;
    no_accept:=0;
    change:=false;
    while ((no_attempt <= max_attempt) or (no_accept <=max_accept)) and (no_attempt
      <= max_attempt*at) do swap;
    if (no_attempt>(max_attempt*at)) and (no_accept<min_accept) and (not change)
      then stop1:= stop1+1 else stop1:=0;
    if (not change) and (no_attempt>max_attempt*at) then stop2:=stop2+1 else stop2:=0;
    no_temp_step:=no_temp_step+1;
    ttt:=ttt*fact;
  until (no_accept=0) or (stop1=max_stop1) or (stop2=max_stop2);
  if no_temp_step <= min_step then begin init.me:=initime*init; goto 999; end;
  clockoff;
end;
```