

## Modified Simulated Annealing을 이용한 다품종 회분식 조업의 생산계획

정재학\* · 원기훈 · 정준영\*\* · 양대륙 · 이인범†

\*영남대학교 공과대학 화학공학과  
†포항공과대학교/공정산업의 지능자동화연구센터  
\*\*호남정유 기술연구소  
(1993년 10월 7일 접수, 1994년 9월 3일 채택)

## Scheduling of Multi-product Batch Processes Using Modified Simulated Annealing

Jae Hak Jung\*, Keehoon Won, June Young Jung\*\*, Dae Ryook Yang and In-Beum Lee†

\*Dept. of Chem. Eng., College of Eng., Yeungnam University  
Automation Research Center/POSTECH

\*\*Research and Development Center, Honam Oil Refinery Co., Ltd.  
(Received 7 October 1993; accepted 3 September 1994)

### 요 약

다품종용 회분식 화학공정이 산업에서 요구되어짐에 따라 다품종 회분식 공정의 최적 생산계획 연구도 활발히 진행되어 왔다. 이러한 생산계획 문제는 NP-complete로 알려져 있을 만큼 정확한 최적해를 구하기가 매우 어려우며 특히 문제의 크기(제품 및 장치의 수)가 커질수록 최적해를 구하기는 더욱 어려워진다. 그래서 최근 생산계획문제의 최적해를 구하는 방법대신 준최적해를 구하는 방법이 연구되어져, 보다 과학적이고 치밀하여진 기법을 통해 최적해에 가까운 해를 얻어내고 있다. 1980년대 말까지는 주로 BAB(branch and bound)법이나 경험적 법칙이 주종을 이루었으나, 최근 Ku와 Karimi[9]는 강철을 더 강하게 만드는 담금질(annealing)의 물리계를 모방하여 만든 Simulated Annealing(SA)법을 생산계획문제에 도입하여 매우 우수한 결과를 얻어 문헌을 통해 밝힌 바 있다. 그러나 생산계획문제를 위한 SA법의 단순도입은 개선의 여지가 많은 것으로 나타났다. 본 연구에서는 이 SA법의 한계점을 극복하기 위해, 경험적 법칙으로 잘 알려진 Rapid Access Extensive Search(RAES)법과 조절변수를 적절히 조작한 Metropolis Simulated Annealing법의 혼합으로 이루어진 Modified Simulated Annealing(MSA)법을 개발하였다. MSA법은 2단계 접근법으로, 첫 단계에서 RAES법을 이용한 보다 우수한 초기점을 얻어낸 후 두번째 단계에서 SA법의 조절변수 조작을 통해 불필요한 영역의 search를 줄여주는 합리적인 annealing simulation을 수행한다. 끝으로 기존 문헌에 나타났거나, 무작위로 만들어진 매우 다양한 문제를 적용하여 각각 풀이해 본 결과, MSA법이 SA법보다 우수함을 보였다.

**Abstract**—As a trend toward multi-product batch processes is increasing in Chemical Process Industry (CPI), multi-product batch scheduling has been actively studied. But the optimal production scheduling problems for multi-product batch processes are known as NP-complete problems. So the methods which can obtain not exact optimal solution but near optimal solution have been studied. Heuristic methods have been used frequently for near optimal solutions but recently Ku and Karimi[9] have studied the Simulated Annealing

ling(SA) which acquires outperformed results in scheduling problems. But SA method can be improved for better results of batch process scheduling problems. In this paper, we suggest the Modified Simulated Annealing(MSA) which is composed of two stage search algorithms. At the first stage, Rapid Access Extensive Search(RAES) algorithm was used for better location of initial state. At the second stage, a modified Metropolis algorithm with adjusted control parameter was developed and used for better solution. A variety of examples are tested to evaluate the performance of MSA and to be compared with the SA method. From these tests we can show that the MSA is superior to the SA.

## 1. 서 론

과거의 화학공정은 대규모 연속식 생산공정으로써의 대량 생산 체계를 지향하여 왔으나, 최근 제품의 양보다는 질이 산업의 경쟁에서 이길 수 있는 길임이 부각되기 시작하였으며, 또한 소비 형태나 소비자의 취향이 매우 다양해지면서 점점 짧은 시간 내에 변동하는 수요에 대응할 수 있는 다품종, 고부가가치, 소량 생산의 회분식 화학공정이 점차 각광을 받기 시작하였다. 이러한 회분식 조업이 관심의 대상이 됨에 따라 연속 공정에서와는 달리 회분식 공정에서 일어날 수 있는 여러 가지 문제점들의 연구가 시작되었다. 그 중에서도 가장 먼저 연구되어져 온 것이 주어진 생산량을 최소의 시간에 생산할 수 있는 생산순서를 얻어내는 연구 즉 생산계획(production sequencing and scheduling)이다. 생산계획 문제는 생산될 제품들이 원료로부터 제품까지의 제조과정이 모두 일정하게 주어진 다품종용(multi-product) flowshop문제와 생산순서가 각 제품별로 다를 수 있거나 일정치 않은 다목적용(multi-purpose) job-shop문제로 나눌 수 있다. 본 연구는 이들 중 다품종용 최적 생산계획 문제로 그 연구방향을 설정하였다.

화학공학에서는 지난 10년간 화학산업을 위한 다품종용 회분식 조업의 생산계획에 관한 많은 연구가 이루어져 왔다. 회분식조업에서 생산효율의 증대방안에 관한 기술 중 하나가 각 회분식 장치들 사이에 중간 저장탱크를 두어 한 장치에서 병목현상(bottleneck)이 생길 때, 그 장치 이전의 모든 장치에 생기는 조업대기현상을 완화시키는 것이다. 이런 중간 저장탱크를 사용하는 혹은 저장탱크없이 조업하는 생산활동의 특징에 따라 여러 가지 중간 저장탱크 사용방안으로 조업을 정의하였다. 각 장치사이에 중간 저장탱크가 필요한 만큼 충분히 존재한다고 가정한 것이 UIS(unlimited intermediate storage) 방안이며, 특정한 한정적인 장치와 장치사이에 한정된 수의 중간 저장탱크가 존재한다고 가정된 것이 FIS(finite intermediate storage) 방안이며, 또 중간 저장탱크 없이 회분식 조업을 진행하는데, 만약 앞의 장치가 조업 중일 경우 그 장치는 조업이 끝났더라도 중간제품은 장치가 그대로 가지고

있어야(hold) 하는 방안이 NIS(no intermediate storage)방안이며, 중간 저장탱크없이 조업을 하지만 장치가 주어진 조업을 완성한 후 지체없이 반드시 다음 장치로 보내어져야 하는 경우의 생산 방안을 ZW(zero wait) 방안이라 한다. 또 다단계 장치의 회분식 조업에서 각 단계의 중간 저장탱크가 모두 한 가지의 저장탱크 사용방안으로 통일되어 있지 않고 각 단계에서 서로 다르거나 적어도 두 가지 이상의 방안을 사용하게 되는 생산방안을 MIS(mixed intermediate storage) 방안이라고 한다. 최근 MIS 방안의 저장탱크를 수 개의 장치가 공유(shared storage)하는 방안이 소개되어 있다. 이들 각각의 저장탱크 사용방안에서의 생산은 고유한 특징이 있어서 한 가지의 생산계획의 방법이 모두에게 잘 적용되어져 좋은 해를 구해주지는 않는다. 생산공정이 다양한 만큼 생산계획법도, 그 공정의 특성에 잘 맞는 것을 적용해야 보다 우수한 결과를 얻을 수 있다. 생산계획법은 이러한 관점에서 다양한 방법으로 개발되어지고 있다. 또한 컴퓨터의 성능발전은 1980년대에 접어들면서 태동되기 시작하여 1980년대 말부터는 엄청난 속도로 발전하여 왔다. 회분식 공정의 다품종 생산계획 문제는 최적해답을 얻기에는 너무나 방대하고 어려운 문제임이 증명되었으며 '80년대 이전의 연구들은 컴퓨터의 도움을 얻을 수 없었기 때문에 앞서 본 바와 같이 거의가 경험적 법칙 개발에 의존하여 왔다. 그러나 1980년대에 들어와 컴퓨터의 도움으로 최적 생산계획 해법을 찾고자 하는 연구들이 많이 이루어지기 시작하였다. 화학공학분야에서 생산계획 문제를 화학공학 시스템에 적용한 것도 1980년 초부터 시작되었으며, 그 후 많은 연구가 이루어져 그동안 Reklaitis[1,2], Ku 등[3]에 의해 여러 차례 review paper가 나오기까지 하였다.

현재까지 개발되어 발표되어진 생산계획의 방법론은 다음과 같은 것들이 주류를 이룬다. 공정을 수학적으로 정확히 표현한 후 최적화 문제를 풀이하는 최적화 기법이 있으며, 경험적 법칙을 이용하여 비교적 단순하고 간단한 방법으로 준 최적 해답을 얻는 방법, BAB(branched and bound) search법, 모든 가능성을 다 조사하는 combinatorial exhaustive search법 그리고 인공

지능 응용법 등이 있고 최근 시행착오적 방법의 단점을 보완한 Simulated Annealing(SA)기법을 이용한 최적화법 등이 연구되어져 있다. 이들은 모두 장단점이 있다. 수학적 최적화 기법이나 combinatorial exhaustive search법은 항상 최적해를 보장하지만 다룰 수 있는 문제의 크기가 매우 제한적이어서 현재 제품수 10개가 넘는 크기의 문제는 풀이가 불가능하다. 반면 경험적 법칙, BAB법, SA법 등은 그 방법론이 다양하게 개발되어 보다 큰 문제를 다룰 수 있게 되었지만 반면 해답의 최적성은 보장할 수 없는 소위 near optimum solution법에 속한다. 이들 중 SA법은 Ku와 Karimi[9]가 적용하여 문헌을 통해 발표하였는데, 그 문헌에서 경험적 법칙 중 가장 우수한 방법 중 하나라고 알려진 IMS(idle matrix search)법에 비해 SA법이 보다 더 우수함을 역설하고 있으며, 또한 상당히 큰 문제도 다룰 수 있는 방법임을 입증하고 있다.

본 연구에서는 이러한 Ku와 Karimi[9]가 제안한 SA법에 비해 보다 우수한 성능을 보이는 Modified Simulated Annealing(MSA)법을 개발하였으며, 무작위로 만들어진 다양한 문제에서 경험적 법칙 중 하나인 RAES법 및 SA법과의 비교를 통해 MSA법의 우수성을 입증하였다.

## 2. 이론적 배경

Johnson[4]은 flowshop생산형태의 공정에서, 장치가 2개인 경우 다품종의 최적 생산계획을 얻어낼 수 있는 알고리즘을 개발하였으며, 그 후 장치가 3개 이상인 경우 최적생산은 얻기 힘들지만, 매우 양호한 생산계획을 얻어낼 수 있는 경험적 법칙을 이용한 방법들이 최근까지 연구되어지고 있다[5-8]. 그 중 Dennenbring[6]은 장치 2개의 Johnson 알고리즘을 토대로 여러 개 장치의 회분식 생산에서의 경험적 법칙들을 다수 개발하여 그들 중 가장 우수한 RAES(rapid access extensive search)법을 제시하였다. 이 RAES법은 다품종, 3장치 이상의 생산공정에서 양호한 준최적 생산계획을 얻어낼 수 있는 대표적인 방법으로 받아들여졌으며, 최근까지도 새로운 경험적 방법의 성능평가 비교대상으로 꾸준히 사용되어지고 있다. 그러나 연구되어진 이들 경험적 방법들은 거의 2장치 Johnson 알고리즘에 그 기본을 두어 조금씩 변형되거나, 전처리 단계를 거치는 방법이 주류를 이루고 있다.

최근 NP-complete한 combinatorial 최적화 문제들의 해법으로 많이 거론되고 있는 것이 SA에 의한 최적화법이다. 기존의 여러 가지 경험적 법칙에 의한 최적화 기법은 대부분 local minimum에 중간결과가 놓여질

경우 더 이상의 search를 하지 않고 local minimum을 해답으로 받아들이게 된다. 그렇기 때문에 몇몇 문제에서는 값을 알지 못하는 최적치와 비교적 많은 차이가 나는 해답을 구하고도 더 이상의 문제풀이를 못하고 나쁜 결과를 도출하기도 하는 경우가 빈번하였다. SA 기법은 이러한 단점을 보완한 것으로써 최적 해답의 search 중 현재의 해답보다 다소 나쁜 해답이 얻어지더라도 문제의 특성상 신빙성있는 확률적 판단에 의해 그 해답을 수용하여, 계속 해답 search를 할 것인지, 그 해답을 버릴 것인지를 결정하게 해주는 알고리즘이다. 이렇게 함으로써 local minimum에서 빠져 나와 보다 더 나은 해답을 얻을 수 있는 확률적 기회가 부여되는 것이다. 이 해법은 Kirkpatrick(1983)에 의해 처음으로 traveling salesman problem에 처음 적용되었으며 화학공학에는 Doln 등(1989)에 의해 열교환망 합성을 위한 최소 설계 비용을 목적함수로 응용되어 우수한 결과를 얻어내었다. 최근 Ku와 Karimi[9]는 회분식 다품종생산의 생산계획 중 가장 간단한 UIS방안에 이 방법을 적용하여 우수한 결과를 얻었음을 문헌을 통해 발표한 바 있다. SA법은 비교적 그 알고리즘이 간단하며, 상당히 큰 문제도 적절한 컴퓨터 계산시간 내에 해답을 얻어내고 있다. 특히 회분식 공정의 다품종생산용 생산계획과 같은 NP-complete한 문제로 알려진 문제들에 그 적용성이 좋고 비교적 우수한 결과가 도출되므로 많이 응용되어 지리라 보여진다. 이러한 SA법은 이미 많은 사람들에 의해 여러 가지 방법이 제시되어져 왔는데, Ku와 Karimi[9]는 이들 중 가장 일반적으로 사용되어지고 있는 Metropolis 알고리즘을 적용하여 생산계획 문제를 풀이하였다. 그 결과 매우 우수한 결과가 도출되기는 하였으나 아직 최적해에 근접하지 못하는 경우도 발생되고 있는 단점을 내포하고 있다.

Metropolis 알고리즘은 시행 착오적 방법에 의해 얻어지는 각각의 해답을 물리적 계의 여러 다른 분자 배열들로 보고, 목적함수값을 물리계의 다양한 에너지 상태식으로 관련지어 보다 나은 분자배열 즉 시행착오적 해답을 얻자는데 그 기본 개념이 있다. 이런 물리계의 현상을 이용한 컴퓨터 알고리즘을 최적화 특히 생산계획에 적용시키기 위해 목적 함수인 에너지 함수를 총 조업시간(makespan)으로 보아야 한다. 에너지 함수값을 초기에 무작위 변수 고정에 의해 얻고 이를  $E_1$ 이라 하며, 그 다음번 무작위 변수 고정에 의한 에너지 함수값을  $E_2$ 라 한다. 이 때  $E_1 > E_2$ 인 경우 새로운 해답이 보다 더 작은 값이 얻어진 것이므로 최소화를 목적으로 할 경우 새로운 중간 해답으로 받아들여지고,  $E_1 < E_2$ 인 경우는 새로운 해답이 과거의 해답보다 우수한 값이 도출되어지지 않았으므로 버려지는 것이 아니라, 확률적

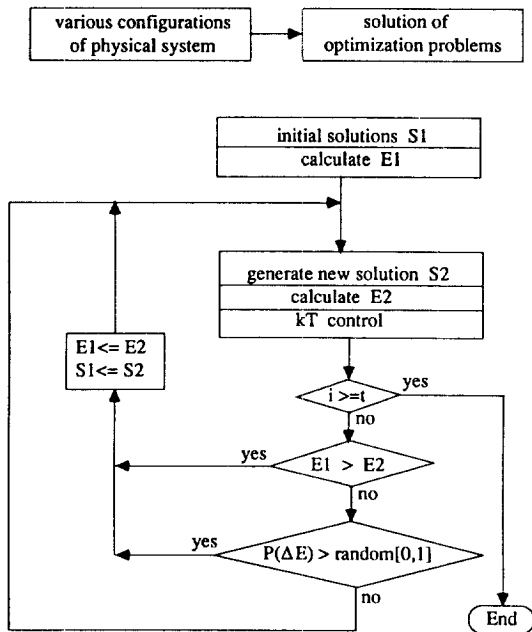


Fig. 1. Flowchart of Metropolis simulated annealing algorithm.

판단에 의해 받아들이거나, 버림을 다시 한번 결정하게 된다. 즉 최소화 문제에 있어서 목적 함수치가 증가하는 경우 받아 들이는 확률을 다음 식 (1)과 같이 두었다.

$$P(\Delta E) = \exp\{-(E_2 - E_1)/k \cdot t\} \quad (1)$$

즉 구간  $[0,1]$ 의 무작위 수(random number)를 발생시켜  $P(\Delta E)$ 와 비교하여  $P(\Delta E)$ 가 발생된 무작위 수보다 큰 경우 새로운 해답을 받아들이고, 작은 경우 새로운 해답을 버리고, 현재의 해답을 계속 유지하면서 다음 단계의 계산을 수행한다. Fig. 1에는 이러한 Metropolis 알고리즘의 논리를 보여주는 흐름도를 나타내었다. 이와 같이 전단계의 해답보다 다음 단계의 해답이 나 빠지더라도 어느 정도의 확률에 의해 그 해답이 받아들여질 수 있는 것을 알 수 있는데 이 확률을 up-hill movement라고 한다. 이 up-hill movement는 식 (1) 중 변수  $t$ 와  $(E_2 - E_1)$ 에 영향을 받아 정해지는데, 이 중  $(E_2 - E_1)$ 은 조정 불가능한 변수이나,  $t$ 값은 조정이 가능하다. 계산 과정이 진행되면서 점점 최적 해답에 가까이 가게 되므로, 과정이 진행되는 동안 확률은 점점 줄어들어야 하는데, 그 확률을 줄여줄 수 있는 조정인자는  $t$ 가 될 수 있다. 최종적으로 이 알고리즘의 종료는 미리 정해진 종료 조건, 즉 주어진 횟수만큼의 시행착오적 해답 계산에 의해 이루어진다.

SA 알고리즘에 의해 최적해를 구하기 위해서는 다음

3 단계의 과정을 거쳐야 한다.

- (1) 목적 함수를 수식화하여 에너지 함수로 삼는다.
- (2) 시행착오적 해답을 위해 무작위 변수를 정의하고, 매번의 무작위 변수 발생시의 에너지 함수값 계산 공식을 도입한다.

(3) Metropolis 알고리즘을 적용한다. 여기에서 중요한 것은 문제의 성격에 맞게 초기의  $E_1$  값을 정하는 방법, 초기의  $T$ 값을 정하는 방법,  $T$ 값을 iteration이 진행되는 동안 줄여 나가는 방법, 프로그램 종료 조건 등이 연구 대상이 될 수 있다.

이밖에 중요한 것으로는 무작위 수(random number)를 효과적으로 발생시키는 방법도 SA법에서는 매우 중요한 요소가 될 수 있다.

Ku와 Karimi[9]는 UIS방안의 조업완료시간  $C_{ij} = \max[C_{(i-1)}, C_{(i-1)j}] + T_{ij}$  ( $i=1, 2, \dots, N, j=1, 2, \dots, M$ )을 이용한,  $C_{NM}$ 의 최소화를 목적 함수로 두었다. 또한 무작위 수를 발생시켜 주어진 제품들의 초기 순서를 결정하였고, 그렇게 얻은 순서에 입각한  $C_{NM}$ 을 조업완료 시간을 나타내는 수학적 회귀공식에 의해 계산하였다. 식 (1)의  $k \cdot t$ 를  $T$ 로 두어, 초기의  $T$ 값은 무작위 수 발생에 의해 제품 순서를 약 3,000회 만들어보고 인접하여 얻어진 조업 순서의  $C_{NM}$ 값들 간의 편차의 절대치 평균값을 계산하고 이 값의 약 1.5배를 곱한 수로 설정했다. 새로운 배열의 총 발생횟수를  $3N^3$ 으로 두고, 이를 20 step으로 나누어 매 step이 넘어 갈 때마다  $T \times 0.95$ 를 하여 새로이 감소되어지는  $T$ 값으로 설정하고 있다.

초기의 조업순서(initial sequence)  $S_1$ 은 0과 1사이의 균등 무작위수 발생(uniform random perturbation)에 의해 무작위로 설정하여 출발한다. 에너지값, 즉 목적 함수  $E_i$ 는 앞서 설명하였듯이 Min.  $C_{MN}$ 으로 두고 up-hill movement의 확률  $P(\Delta E) = \exp\{-(E_i - E_{i-1})/k \cdot t_i\}$ 가 된다.  $k$ 는 boltzman 상수로 여기에서는  $k \cdot t_i = T_i$ 로 두어 풀이하게 된다.  $T_1 = k \cdot t_1$ , 즉 초기 조작 변수를  $1.5 \cdot \left( \sum_i |\Delta E_i| \right) / 3000$ 으로 하였고 20 step을 내려가면서 이 값이 조금씩 줄어든다. 즉  $T_j = k \cdot t_j = 0.95 \cdot k \cdot t_{j-1} = 0.95 \cdot T_{j-1}$  ( $j=1, 2, \dots, 20$ )이 되므로 최종의  $T_{20}$ 은  $0.95^{20} \cdot T_1$ 이 되므로  $T_{20}$ 은 약  $0.36 \cdot T_1$ 이 된다. Annealing 시뮬레이션을 진행함에 따라 생성되는 조업순서(updated sequence)  $S_i$ 는 역시 균등 무작위수 발생 원리에 의해 생산제품 순서의 한 위치를 선택하고 그 다음 위치의 제품과 자리바꿈을 통해 일어나는 순서 수정법을 사용하고 마지막으로 프로그램 종료 조건은 iteration수를 제한하여 마무리 된다. 이것은 문제의 크기에 따라 적당히 수정되어야 하겠으나 무엇보다도 비교적 현실적인 컴퓨터 계산 및 사용 시간을 고려하여 정해진다. Ku와

Table 1. Processing times for example 1

Product	Unit 1	Unit 2	Unit 3
1	61	2	56
2	42	32	42
3	1	31	71
4	46	98	42
5	11	14	20
6	35	89	75
7	66	8	25
8	25	46	44

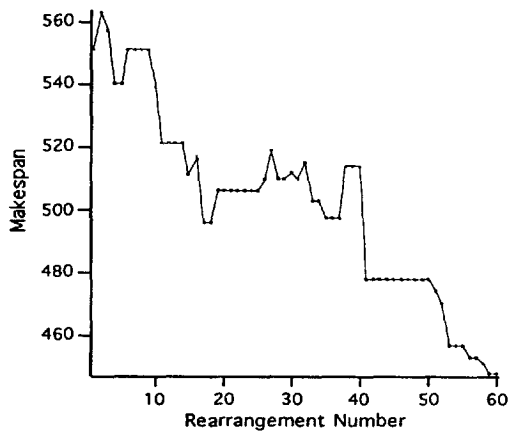


Fig. 2. Performance of simulated annealing algorithm on example 1.

Karimi[9]는 이것을 제품수의 3제곱의 3배( $3N^3$ )으로 두었다. Ku와 Karimi[9]는 그들의 SA법을 다음 Table 1에 나타난 문제에 적용시켜 약 60번의 iteration 동안 보다 나은 해답을 찾아가는 경로를 나타냄으로써 SA법이 매우 유효적절한 방법임을 보여주고 있다. 그 경로는 Fig. 2에 나타내었다.

### 3. Modified Simulated Annealing 법

앞서 설명한 대로 현재까지의 다품종용 회분식 공정의 생산계획 문제에 관한 연구 방법들이 다양하게 문헌을 통해 소개되고 있으나, 그 성능의 절대적 평가를 한마디로 단언할 수는 없는 실정이다. 모든 방법들이 장단점을 갖고 있고, 또한 부대적으로 컴퓨터 시스템의 발달은 매우 급속히 발전하고 있어, 각 연구 방법들에 대한 평가도 날로 새로와 질 것이다. 과거 10여년 전에는 컴퓨터 풀이 시간이 매우 중요하여 경험적 법칙이 가장 많이 연구되어져 왔고, 좋은 평가를 받아 와 문헌에 소개되는 빈도가 가장 높았다. 그러나 최근 컴퓨터 성

능이 눈부시게 발전함에 따라 이러한 컴퓨터의 능력을 빌려 수행되는 연구가 각광받기 시작하였고, 이 중 Monte-Carlo 시뮬레이션 기법을 응용한 Random Search법이 부각되고, 그 성능이 입증되고 있다.

이러한 관점에서 Ku와 Karimi[9] 접근법은 매우 유용한 것이라 평가되고 있다. 그러나 Ku와 Karimi의 접근법은 아직도 다소 개선해야 할 점이 보여지고 있는바, 본 연구에서는 이들 방법의 단점을 보완하여 같은 컴퓨터 계산 시간 내에 보다 우수한 성능을 지닌 새로운 방법을 연구하게 되었다. Ku와 Karimi의 방법은 기존의 Metropolis 알고리즘에 바탕을 두어, 매우 기본적인 SA기법을 그대로 생산계획의 최적화 문제에 적용한 것이라 말할 수 있다. 다만 조절변수 채택, 프로그램 종료조건 (terminal condition), up-hill movement의 확률 등 구체적인 수치를 생산계획의 최적화에 맞게 얻어내어 제시하는데 그 의의가 있다. 그렇기 때문에 초기의 생산순서를 무작위 생산품 배열로부터 출발하고 있고, 이렇게 되면 매우 좋은 생산계획이거나, 매우 나쁜 생산계획이 확률보다는 정규분포상의 가정하에 평균정도의 우수한 결과로 출발하게 될 확률이 가장 클 것이다. 그렇기 때문에 초기의 up-hill movement 확률을 다음 식 (2)로 두었다.

$$\begin{aligned}
 P_1(\Delta E) &= \exp\{-(E_2 - E_1)/k \cdot t_1\} \\
 &= \exp\{-(E_2 - E_1)/1.5 \cdot \left(\sum_i^{3000} |\Delta E_i|\right)/3000\} \\
 &\sim 0.513
 \end{aligned} \quad (2)$$

만약  $(E_2 - E_1)$ 이  $\left(\sum_i^{3000} |\Delta E_i|\right)/3000$ 에 가까울 경우  $P_1(\Delta E)$ 의 값인 up-hill movement의 확률이 0.513 즉, 약 반 정도로 출발하도록 하고 있다. 또한 최종의 확률은 다음 식 (3)으로 하고 있다.

$$\begin{aligned}
 P_{20}(\Delta E) &= 0.36 \cdot \exp\{-(E_i - E_{i-1})/k \cdot t_i\} \\
 &= 0.36 \cdot \exp\{-(E_i - E_{i-1})/1.5 \cdot \left(\sum_i^{3000} |\Delta E_i|\right) / 3000\} \sim 0.185
 \end{aligned} \quad (3)$$

하지만 회분식 다품종용의 생산계획 연구에서 자주 거론되는 경험적 법칙들을 통해 볼 때 특정의 결과가 나쁜 제품생산순서들은 그들의 이웃한 제품 순서들에서도 좋은 해답을 기대하기 어렵다는 결론을 이끌어 내고 있다. 즉 좋은 해답들은 경험적 법칙에 의해 얻어진 근처에 많이 분포하고, 또 전체의 search 공간 중 좋은 해답의 가능성이 거의 없는 특정 부분을 미리 알 수 있어 그 부분의 search는 불필요함을 단정지을 수 있는 경우가 많다.

본 연구에서는 이러한 관점에서 보다 효율적으로 최적해를 구할 수 있는 접근법을 연구하였다. 기존의 Metropolis 알고리즘이 이러한 특징을 갖고 있는 다품종용 회분식 공정의 생산계획 문제에서 더욱 효율적이기 위해, 방대한 search 공간을 줄여 주어 보다 최적해를 얻을 가능성이 높은 공간에서 보다 조밀한 search를 행함으로써 숨어있는 더 좋은 해답을 이끌어 내어, 같은 컴퓨터 계산 시간으로 보다 나은 답을 얻고자 하였다. 그러기 위하여 우선 해답이 숨어 있을 가능성이 적은 지역을 피하여 search의 출발을 시도하여야 하며, 이러한 해법이 진행되는 동안 해답이 있을 가능성이 없는 지역으로의 탈출을 적절히 막아주어야 한다.

기존의 단순 Metropolis 알고리즘에 의한 SA법이 회분식 다품종용 최적 생산계획에 보다 효율적으로 적용되도록 만든 본 연구의 방법을 Modified Simulated Annealing(MSA)이라고 명명하였다. MSA법은 다음의 2가지 단계로 이루어진다. 첫째 단계에서는 보다 좋은 해답이 많고, 최적해가 숨어있을 가능성이 많은 지역에서의 초기 search를 위해 매우 계산이 간단하고 그 우수성이 상대적으로 인정된 RAES법을 사용한다. RAES법은 회분식 flowshop, UIS 중간저장탱크의 문제에서 비교적 작은 제품수 즉 10개 이하의 문제에서 약 10% 이내의 최적 해답을 얻을 수 있다고 보고되고 있고 제품수 약 20-30개 이상의 큰 문제에서도 약 20% 이내의 해답을 나타낼 수 있는 방법으로 알려지고 있다. 뿐만 아니고, 양호한 해답이나 최적해가 RAES나 그 밖의 경험적 방법에 의해 반드시 찾아지는 것은 아니지만, 그 해답 주변에 있을 확률이 많고, 상대적으로 그 해답과 동떨어진 곳에서 우수한 해답이 도출될 확률은 작다고 할 수 있다. 결국 이러한 양호한 해답에서 출발하였기 때문에 초기의 up-hill movement의 확률을 단순 SA법에서와 같은  $P_1(\Delta E) \sim 0.5$  정도의 값을 쓸 수는 없다. 이것은 이런 양호한 해답의 구역을 포기하고자 하는 경향을 다소 크게 주는 결과가 되기 때문이다. 그래서 두번째 단계에서는 초기의 up-hill movement의 확률을 보다 작게 주어서 상당히 우수한 새로운 해답이 아닐 경우 그 해답을 받아들이는 확률을 보다 tight하게 하고 있다. Metropolis 알고리즘에 그 근간을 둔 random search의 성능은 크게 다음의 3가지에 달려 있다. 첫째 무작위수 발생기(random number generator)의 성능이며, 둘째 초기상태(initial state)이며, 마지막으로 에너지 함수의 up-hill movement 확률값  $P(\Delta E)$ 이다. 이외에도 확률의 줄임폭, 새로운 제품생산순서 생성의 법칙, 프로그램 종료 조건 등이 다소 영향을 준다. 그러나 방법론의 비교를 위해 무작위수 발생기는 같은 것으로 통일하여 사용하였다. 본 연구에서 사용한 MSA

법은 결국 기존의 경험적 법칙과 다소 수정된 Metropolis 알고리즘의 복합적 방법으로 초기의 상태를 RAES 경험법칙을 통해 얻어내고, 초기의 up-hill movement 확률을 UIS 중간저장탱크의 문제에서는 식 (4)로 둔다.

$$\begin{aligned} P_1(\Delta E) &= \exp\{-(E_2 - E_1)/k \cdot t_1\} \\ &= \exp\{-(E_2 - E_1)/0.75 \cdot \left(\sum_i^{3000} |\Delta E_i|/3000\right)\} \\ &\sim 0.264 \end{aligned} \quad (4)$$

즉 초기 확률  $P_1(\Delta E)$ 를 Ku와 Karimi[9]의 약 반정도를 설정해  $-(E_1 - E_2)$  값의 크기에 따라 다르겠으나, 평균  $\Delta E$ 값과 비슷하다면 받아들여질 확률이 약 1/4 정도로 보다 엄격하게 출발하고 있다. 최종 확률은 식 (5)가 되어 약 10% 이하가 된다.

$$\begin{aligned} P_{20}(\Delta E) &= 0.36 \cdot \exp\{-(E_i - E_{i-1})/k \cdot t_i\} \\ &= 0.36 \cdot \exp\{-(E_i - E_{i-1})/0.75 \cdot \left(\sum_i^{3000} |\Delta E_i|/3000\right)\} \\ &\sim 0.095 \end{aligned} \quad (5)$$

이러한 방법을 보다 광범위하게 적용해 보기 위해 본 연구에서는 NIS 중간저장탱크에 관해서도 그 적용법을 연구하였다. RAES 경험적 법칙은 원래 UIS 중간저장탱크를 전제로 개발된 2개의 장치 Johnson법칙[4]을 다수의 장치에서도 적용할 수 있도록 하기 위해 만들어진 것이기 때문에 FIS, NIS, ZW, MIS 등의 다양한 중간저장탱크 옵션에서는 다소 덜 우수한 결과를 나타내어 주고 있다. 이들 중 UIS와 가장 관계가 적은(즉 중간저장탱크 사용의 차이가 가장 큰) NIS 중간저장탱크에서도 본 연구의 방법인 MSA가 매우 잘 적용되게 하기 위한 새로운 조작 변수를 얻어 내었다. RAES 경험적 법칙이 UIS 옵션의 경우 최적 해답의 10-20%에 들어 오는 방법임을 앞서 밝힌 바 있는데, NIS 옵션의 경우는 최적 해답 보다 다소 떨어진 해답이 존재할 것이므로 up-hill movement의 확률을 조금 크게 조절해 줄 필요가 있다. 결국 초기의 up-hill movement의 확률은 NIS 중간저장탱크의 문제에서는 식 (6)으로 한다.

$$\begin{aligned} P_1(\Delta E) &= \exp\{-(E_2 - E_1)/k \cdot t_1\} \\ &= \exp\{-(E_2 - E_1)/0.9 \cdot \left(\sum_i^{3000} |\Delta E_i|/3000\right)\} \\ &\sim 0.329 \end{aligned} \quad (6)$$

초기 up-hill movement의 확률이 0.329 즉 약 1/3 정도로 두는 결과가 되며 최종의 확률은 다음 식 (7)과 같이 된다.

$$P_{20}(\Delta E) = 0.36 \cdot \exp\{-(E_i - E_{i-1})/k \cdot t_i\}$$

$$= 0.36 \cdot \exp\left\{-\left(E_i - E_{i-1}\right) / 0.9 \cdot \left(\sum_i^{3000} |\Delta E_i| \right) / 3000\right\} \sim 0.119 \quad (7)$$

$$C_{ij}(\text{UIS}) = \max[C_{(i-1)j}, C_{ij-1}] + t_{ij} \quad (8)$$

$$C_{ij}(\text{NIS}) = \max[C_{(i-1)j}, C_{ij-1}, C_{(i-1)(j+1)} - t_{ij}] + t_{ij} \quad (9)$$

목적함수로 사용할 UIS 옵션 및 NIS 옵션의 다품종 회분식 조업의 완료시간 결정을 위한 회귀적 수학적 공식(recurrence relation)은 Ku 등[3]과 Ku와 Karimi[11]에 나타나 있는 다음 식 (8)과 (9)를 이용하였다.

#### 4. 예제 풀이 및 결과

본 연구에서 다루어진 문제는 우선 기존의 문헌을

Table 2. The result and evaluation for small size problems with UIS policy

# of prob.	Size	Optimal	# of opt.	% of opt.	RAES		SA		MSA	
					sol.	dist. %	sol.	dist. %	sol.	dist. %
1	6×3	29	1	0.14	29	0.0	29	0.0	29	0.0
2	6×3	100	2	0.28	102	2.0	100	0.0	100	0.0
3	6×5	138	1	0.14	138	0.0	138	0.0	138	0.0
4	6×5	295	3	0.42	305	3.4	295	0.0	295	0.0
5	6×8	192	1	0.14	192	0.0	192	0.0	192	0.0
6	6×8	274	1	0.14	281	2.6	274	0.0	274	0.0
7	6×10	224	1	0.14	229	2.2	224	0.0	224	0.0
8	6×10	452	2	0.28	469	3.8	452	0.0	452	0.0
9	7×3	70	13	0.26	74	5.7	70	0.0	70	0.0
10	7×3	104	8	0.16	105	1.0	104	0.0	104	0.0
11	7×5	150	12	0.24	150	0.0	150	0.0	150	0.0
12	7×5	234	1	0.02	243	3.9	237	1.3	237	1.3
13	7×8	177	12	0.24	183	3.4	177	0.0	177	0.0
14	7×8	450	7	0.14	479	6.4	450	0.0	450	0.0
15	7×10	256	1	0.02	256	0.0	257	0.4	256	0.0
16	7×10	454	1	0.02	502	10.6	456	0.4	454	0.0
17	8×3	84	7	0.02	92	9.5	84	0.0	84	0.0
18	8×3	411	3	0.007	411	0.0	411	0.0	411	0.0
19	8×5	66	4	0.01	67	1.5	66	0.0	66	0.0
20	8×5	284	6	0.02	290	2.1	289	1.8	284	0.0
21	8×8	118	6	0.02	119	0.9	118	0.0	119	0.9
22	8×8	229	78	0.19	241	5.2	229	0.0	229	0.0
23	8×10	155	2	0.005	165	6.5	157	1.3	155	0.0
24	8×10	456	3	0.007	464	1.8	456	0.0	456	0.0
25	10×3	146	84	0.002	147	0.7	146	0.0	146	0.0
26	10×3	338	41	0.001	342	1.2	341	0.9	338	0.0
27	10×5	176	8	-	184	4.6	176	0.0	176	0.0
28	10×5	318	3	-	330	3.8	324	1.9	319	0.3
29	10×8	202	35	0.001	209	3.5	205	1.5	202	0.0
30	10×8	289	2	-	315	9.0	294	1.7	289	0.0
31	10×10	243	8	-	248	2.1	248	2.1	243	0.0
32	10×10	371	4	-	396	6.7	378	1.9	374	0.8
probability of optimal solution(%)					18.8		65.6		87.5	
mean distance from optimal(%)					3.3		0.5		0.1	
max. distance from optimal(%)					10.6		1.9		1.3	

SA(Simulated Annealing), MSA(Modified Simulated Annealing)

$$\text{distance from optimal(}\%) = \left( \frac{\text{solution of each method} - \text{optimal solution}}{\text{optimal solution}} \times 100 \right)$$

통해 발표되었던[3, 9, 11] 문제들을 사용하였고, 그 외의 문제들은 무작위로 만들어진 문제들을 풀이하였다. 문제의 크기는 제품수 N이 6, 7, 8, 10의 비교적 작은 문제와 15, 20, 30의 큰 문제들로 구성되며, unit 수는 각 문제에 대해 3개, 5개, 8개, 10개를 각각 풀이하였다. 같은 크기(N×M)의 문제도 두 종류로 만들었는데, 비교적 각 unit에서 소요되는 조업시간들의 편차가 다소 큰 문제와 작은 문제들이다. 총 문제수는 56문제가 되고 이들 문제들을 각각의 방법으로(RAES, SA, MSA) 풀이하여 그 성능을 비교하여 보았다.

각 방법들의 성능을 분석하기 위해 작은 문제들의 경우 최적 해답을 exhaustive search로 최적해를 얻어 그 최적해를 기준으로 삼았으며, 최적해의 접근 정도를 나타내기 위해 다음의 식 (10)을 사용하여 분석하였다.

$$\frac{(\text{solution of each method} - \text{optimal solution})}{\text{optimal solution}} \times 100 \quad (10)$$

큰 문제들의 경우는 최적해를 알 수 없으므로 기준치를 RAES 해로 두었고, 각 방법의 해답과 차이를 다음 식 (11)을 통해 향상도 %로 나타내고 있다.

$$\frac{(\text{solution of RAES} - \text{solution of each method})}{\text{solution of RAES}} \times 100 \quad (11)$$

결국 작은 문제 분석을 위한 최적해 접근 정도는 항상 0 이상의 수치가 도출되고 그 값이 0에 가까울수록 좋은 결과임을 알 수 있으며, 큰 문제 분석을 위한 향상도 %값은 양수 혹은 음수로 도출될 수 있으며 보다 큰 값을 나타내는 방법이 향상도가 그만큼 우수한 것임을 나타낸다. 각 문제 풀이에 대한 결과가 UIS인 경우 작은 문제에 대해 Table 2에 나타내었고, 큰 문제에 대해 Table 3에 나타내었다. 또한 NIS의 경우도 마찬가지로 작은 문제의 경우 Table 4에 나타내었고, 큰 문제의 경우 Table 5에 각각 나타내었다. 이 결과들을 Fig. 3-6에 각각 막대 그래프로 나타내었다.

Table 2-5와 Fig. 3-6에 나타난 결과를 볼 때 우선 UIS 옵션의 문제의 경우 본 연구에서 제안한 방법인 MSA법의 결과가 전범위에 걸쳐 우수하게 나타났다. 본 연구에서 다루었던 문제의 범위 내에서, SA법의 결과에 비해 MSA법의 결과가 더 나쁘게 나타난 경우는 한 문제에 불과하며, 더 우수한 결과가 도출된 경우는 27개의 문제나 되었다. 그외의 문제에서는 같은 해답을 얻어내고 있다. 제품이 10개이하의 작은 문제에서 최적해를 찾아내는 확률(%)이, RAES법의 경우 18.8%, SA법의 경우 65.6%가 되나, MSA법의 경우 87.5%로써

**Table 3. The result and evaluation for large size problems with UIS policy**

# of prob.	Size of prob.	Solution of RAES	Solution of SA	Super. to RAES (%)	Solution of MSA	Super. to RAES (%)
33	15x3	114	114	0.0	114	0.0
34	15x3	230	230	0.0	230	0.0
35	15x5	306	306	0.0	306	0.0
36	15x5	469	458	2.3	443	5.5
37	15x8	156	153	1.9	150	3.9
38	15x8	503	460	8.6	449	10.7
39	15x10	312	298	4.5	289	7.4
40	15x10	373	374	-0.3	360	3.5
41	20x3	164	164	0.0	164	0.0
42	20x3	336	336	0.0	336	0.0
43	20x5	319	309	3.1	306	4.1
44	20x5	411	411	0.0	408	0.7
45	20x8	340	313	7.9	309	9.1
46	20x8	530	472	10.9	472	10.9
47	20x10	377	360	4.5	350	7.2
48	20x10	755	681	9.8	676	10.5
49	30x3	466	466	0.0	466	0.0
50	30x3	1011	998	1.3	998	1.3
51	30x5	526	507	3.6	498	5.3
52	30x5	974	948	2.7	939	3.6
53	30x8	616	574	6.8	568	7.8
54	30x8	1314	1197	8.9	1182	10.1
55	30x10	668	654	2.1	650	2.7
56	30x10	1497	1389	7.2	1376	8.1
mean super. to RAES(%)				3.6		4.7
probability of best sol.(%)				33.3		100.0

SA(Simulated Annealing), MSA(Modified Simulated Annealing)

superiority to RAES(%)

$$= \left( \frac{\text{RAES solution} - \text{solution of each method}}{\text{RAES solution}} \times 100 \right)$$

매우 높은 확률로 최적해를 찾고 있음을 알 수 있다. 또한 각 방법에 의해 구해지는 해답이 최적치와 어느 정도나 벗어난 준 최적치인가를 알기 위해 평균 최적해와의 거리(%)를 알아본 결과, RAES법의 경우 3.3%, SA법의 경우 0.5%, MSA법의 경우 0.1%이고, 가장 나쁜 답의 경우 RAES는 최적치와 약 10.6% 차이의 해답을 보이며, SA법은 1.9%, MSA법은 1.3% 정도의 해답을 나타내고 있다. 앞서 밝힌 바와 같이 RAES법의 최적치와의 오차 허용한계가 약 10%정도임이 여기서도 입증되고 있어서 본 예제풀이 검증의 신뢰성이 높음을 보여준다. 제품 15개 이상의 큰 문제에서는 최적치를



Table 4. The result and evaluation for small size problems with NIS policy.

# of prob.	Size	Optimal	# of opt.	% of opt.	RAES		SA		MSA	
					sol.	dist. %	sol.	dist. %	sol.	dist. %
1	6×3	30	1	0.14	31	3.3	30	0.0	30	0.0
2	6×3	103	1	0.14	108	4.9	103	0.0	103	0.0
3	6×5	140	1	0.14	140	0.0	140	0.0	140	0.0
4	6×5	303	3	0.42	306	1.0	305	0.7	303	0.0
5	6×8	201	1	0.14	207	3.0	201	0.0	201	0.0
6	6×8	275	1	0.14	275	2.6	275	0.0	275	0.0
7	6×10	229	1	0.14	229	0.0	229	0.0	229	0.0
8	6×10	461	1	0.14	469	1.7	461	0.0	461	0.0
9	7×3	77	3	0.06	80	3.9	77	0.0	77	0.0
10	7×3	119	2	0.04	126	5.9	123	3.4	119	0.0
11	7×5	150	1	0.02	150	0.0	150	0.0	150	0.0
12	7×5	252	1	0.02	261	3.6	252	0.0	258	2.4
13	7×8	181	4	0.08	183	1.1	183	1.1	181	0.0
14	7×8	463	1	0.02	485	4.8	463	0.0	463	0.0
15	7×10	271	2	0.04	271	0.0	271	0.0	271	0.0
16	7×10	462	1	0.02	518	12.1	472	2.2	462	0.0
17	8×3	89	1	0.003	95	6.7	90	1.1	91	2.2
18	8×3	452	2	0.005	481	6.4	452	0.0	456	0.9
19	8×5	72	4	0.01	72	0.0	72	0.0	72	0.0
20	8×5	300	1	0.003	314	4.7	302	0.7	303	1.0
21	8×8	122	11	0.03	122	0.0	122	0.0	122	0.0
22	8×8	236	1	0.003	243	3.0	237	0.4	237	0.4
23	8×10	161	1	0.003	170	5.6	164	1.9	165	2.5
24	8×10	463	1	0.003	478	3.2	466	0.7	466	0.7
25	10×3	153	-	-	166	8.5	153	0.0	153	0.0
26	10×3	341	-	-	364	6.8	343	0.6	341	0.0
27	10×5	197	-	-	234	18.8	205	4.1	203	3.1
28	10×5	335	-	-	402	20.0	342	2.1	338	0.9
29	10×8	211	-	-	233	10.4	213	1.0	213	1.0
30	10×8	306	-	-	334	9.2	311	1.6	315	2.9
31	10×10	252	-	-	269	6.8	252	0.0	252	0.0
32	10×10	393	-	-	453	15.3	399	1.5	395	0.5
probability of optimal solution(%)					21.9		53.1		62.5	
mean distance from optimal(%)					5.3		0.7		0.6	
max. distance from optimal(%)					20.5		4.1		3.1	

SA(Simulated Annealing), MSA(Modified Simulated Annealing)

$$\text{distance from optimal(}\%) = \left( \frac{\text{solution of each method} - \text{optimal solution}}{\text{optimal solution}} \times 100 \right)$$

얻을 수 없기 때문에 각 방법에 대한 상대적 우위도를 나타내어 비교하였다. 그러므로 이 비교는 수치가 클수록 우수하다. 즉 수치가 0%이면 RAES법과 동일한 성능을 나타낸다고 볼 수 있고, 수치가 커질수록 RAES법에 비해 그 수치만큼 더 우수한 결과를 보임을 뜻한다. SA의 경우 평균 3.6% 정도 RAES보다 우수하며 MSA는 5.2% 정도 우수한 것으로 나타났다. 또 이들 방법들 중 가장 좋은 해답을 얻어낼 확률(probability of best solu-

tion(%))을 검토해 본 결과 SA는 33.3%이며 MSA는 100%로 나타났다. 최우수해답을 두 방법이 동시에 구할 수 있으므로 확률을 합하면 100%가 넘는 경우가 많다.

NIS의 문제는 UIS 문제의 경우와 다소 다른 결과가 나타났다. NIS 문제에서는, 전범위에 걸쳐 고르게 MSA법이 우수한 것으로 나타난 UIS경우와는 달리, 작은 문제에서는 조금 MSA법이 우수한 것으로 나타났으나 큰 문제로 갈수록 MSA의 우수성이 더욱 크게 나타났다.

**Table 5. The result and evaluation for large size problems with NIS policy**

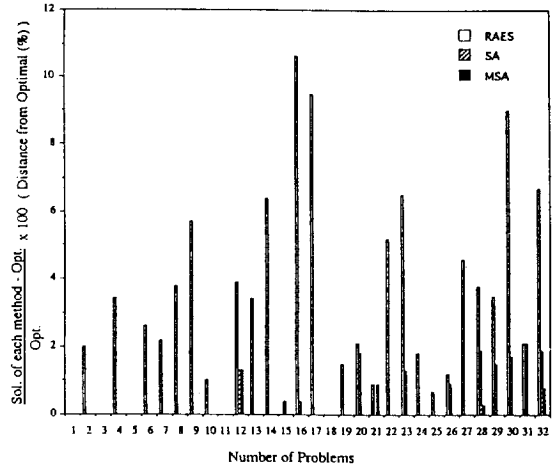
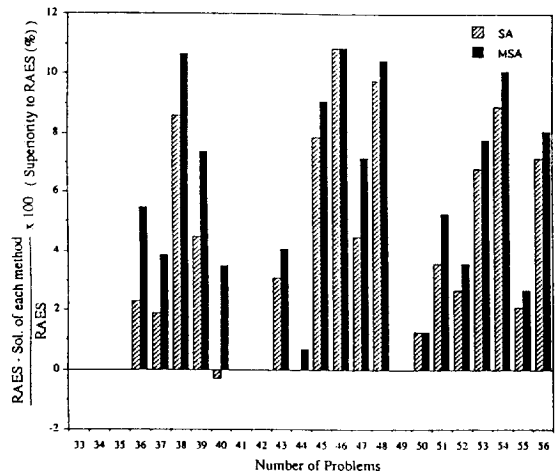
# of prob.	Size of prob.	Solution of RAES	Solution of SA	Super. to RAES (%)	Solution of MSA	Super. to RAES (%)
33	15×3	131	126	3.8	124	5.3
34	15×3	292	262	10.3	258	11.6
35	15×5	369	345	6.5	353	4.3
36	15×5	546	545	0.2	524	4.0
37	15×8	180	172	4.4	173	3.9
38	15×8	576	540	6.3	537	6.8
39	15×10	350	341	2.6	338	3.4
40	15×10	478	451	7.7	451	7.7
41	20×3	201	177	11.9	177	11.9
42	20×3	368	349	5.2	356	3.3
43	20×5	422	381	9.7	377	10.7
44	20×5	834	567	32.0	520	37.7
45	20×8	438	385	12.1	381	13.0
46	20×8	886	636	28.2	631	28.8
47	20×10	466	421	9.7	423	9.2
48	20×10	1024	804	21.5	784	23.4
49	30×3	595	496	16.6	487	18.2
50	30×3	1257	1129	10.2	1117	11.1
51	30×5	748	622	16.9	604	19.3
52	30×5	1300	1151	11.5	1158	10.9
53	30×8	716	682	4.8	655	8.5
54	30×8	1605	1397	13.0	1388	13.5
55	30×10	870	768	11.7	753	13.5
56	30×10	1975	1732	12.7	1636	17.2
mean super. to RAES(%)				11.2	12.4	
probability of best sol.(%)				29.2	79.2	

SA(Simulated Annealing), MSA(Modified Simulated Annealing)

superiority to RAES(%)

$$= \left( \frac{\text{RAES solution} - \text{solution of each method}}{\text{RAES solution}} \times 100 \right)$$

제품수 10이하의 작은 문제에서 최적해를 찾아내는 확률에 있어서 RAES법은 21.9%, SA 법은 53.1%, MSA 법은 62.5%이며, 평균 최적해와의 차이에서도 RAES법은 5.3%, SA법은 0.7%, MSA법은 0.6%로 MSA법이 다소 우수한 결과를 유지하고 있다. 가장 나쁜 해답의 경우 RAES는 20.0% 최적치와의 차이를 보이나, SA법은 4.1%, MSA법은 3.1%를 보이고 있다. 이렇듯 작은 크기의 문제에서는 RAES법에 비해 SA법 및 MSA법이 모두 상당히 우수한 결과를 나타내고 있고 SA법에 비해 MSA법이 우수하다고 단정지을 수 없는 결과가 드러나고 있다. 하지만, 문제의 크기가 비교적 큰, 제품수

**Fig. 3. Diagram of distance from optimal for small size problems with UIS policy.****Fig. 4. Diagram of superiority to RAES for large size problems with UIS policy.**

15개 이상의 문제들에서는 그 격차가 확연히 드러나기 시작한다. RAES법에 비한 상대적 우수도(%) 평가에서 SA법은 평균 8.8% 우수한 것으로 나타났지만, MSA법의 경우 12.4%로 상당히 우수함을 보여주고 있다. 또한 이들 방법 중 가장 우수한 해답을 얻어낼 확률에서도 SA법은 29.2%에 그치는 반면 MSA법은 79.2%로 매우 높다.

이러한 결과를 토대로 할 때 기존의 Ku와 Karimi[9]가 개발한 SA법에 비해 본 연구에서 개발한 경험적 법칙과 modified Metropolis 알고리즘의 2단계 접근법인 MSA법은 보다 우수한 기법이라고 결론지을 수 있다.

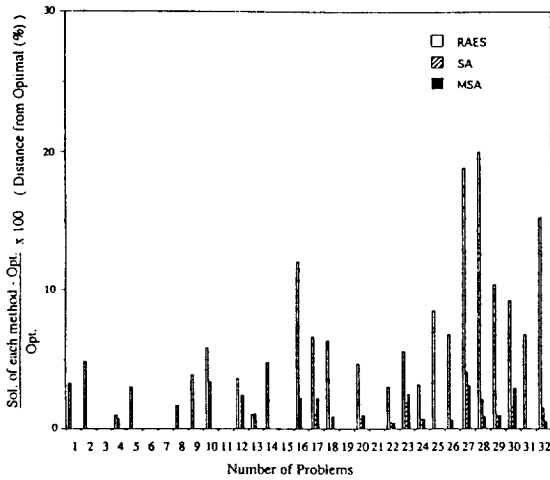


Fig. 5. Diagram of distance from optimal for small size problems with NIS policy.

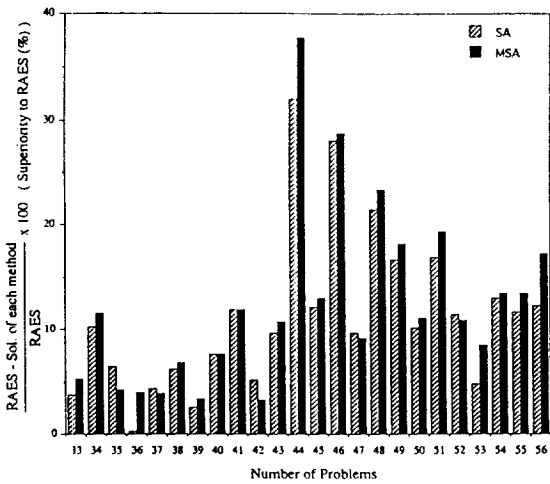


Fig. 6. Diagram of superiority to RAES for large size problems with NIS policy.

컴퓨터 계산시간 면에서도 기존의 SA법과 거의 같고, 문제의 크기에 따라 다소 차이는 있겠으나, IBM호환성 개인용 컴퓨터 486DX/2를 이용하여 DOS상에서 C언어로 프로그래밍을 한 경우 제품수 10개 이하의 작은 문제는 약 10여초 이내이며, 제품수 30개의 큰 문제에서도 약 10여분 이내에 풀이가 끝나기 때문에 매우 실용성이 있는 것으로 평가할 수 있다.

## 5. 결 론

앞의 예제풀이에서 나타난 바와 같이 MSA법을 사

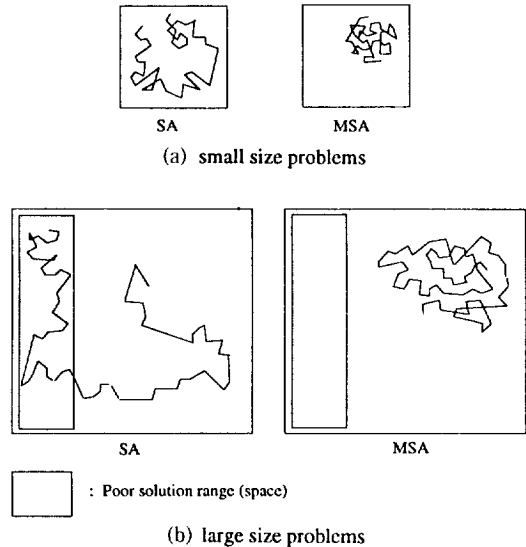


Fig. 7. The conceptual figure of search space and search trajectory.

용한 생산계획의 결과가 전반적으로 우수함을 알 수 있다. Ku와 Karimi[9]에 의하면 SA법은 경험적 법칙들 중 가장 우수하다고 문헌을 통해 발표된 것 중 하나인 Rajagopalan과 Karimi[8]의 IMS(Idle Matrics Search) 법보다 월등히 우수한 결과를 나타내는 방법이다. 그러나, 본 연구에 의한 MSA법은 이러한 SA법에 비해, 작은 문제에서는 그 해답들의 차이가 그다지 크지 않으나, 문제크기가 커질수록 그 결과가 보다 더 우수해지는 경향을 볼 수 있다. 그 이유는 Fig. 7에서와 같이 개념적으로 추측되어질 수 있다. 즉 같은 컴퓨터 계산 시간대 내에서 MSA법이 SA법보다 더 좋은 결과를 나타내고 있는 이유를 개념적으로 분석해 볼 때 다품종 최적 생산계획 문제의 search space의 크기에서 그 원인을 찾을 수 있다. 조업순서가 일정한 다품종 회분식 생산의 조업방법(제품의 조업순서)은  $N!$  ( $N$ =제품수)의 search space를 갖는다. 즉 작은 문제( $N$ 이 약 10 이하)의 경우 총 조업방법의 수는  $10!$  즉 3,628,800가지에 지나지 않으나 큰 문제에서는  $15!$ ,  $20!$ ,  $30!$ 로써 그 search space크기는 천문학적 수치를 나타내게 된다. 제품수가 1개 늘면서 search space는 제품수 즉  $N$ 배씩 커진다. 이러한 이유로 다품종 생산계획 문제를 NP-complete하다고 한다. 그러나 SA법이나 MSA법에서의 search횟수는  $3N^3$ 으로, 문제가 커지면서 증가되는 search space에 비해 매우 조금 증가하므로 보다 효율적인 search방법이 요구되어 진다. Fig. 7에서 나타나 있듯이 작은 문제의 search space에서는 짧은 시간 내에 전

범위에 걸쳐 보다 나은 값을 찾을 수 있으나, 큰 문제의 경우 상당한 시간의 컴퓨터 계산 후에도 겨우 search space의 일부분만을 찾는 정도에 불과할 뿐이다. 그렇기 때문에 RAES 경험법칙을 초기에 이용하여 상당히 우수한 결과가 모여있을 가능성이 높은 지역을 찾아내어, 그 지역을 중심으로 annealing 시뮬레이션을 행하는 MSA법이 기존의 SA법보다 우수한 결과를 얻을 수 있을 것이다. 생산계획 문제의 특징 중 하나가(많은 경험적 법칙들에서 자주 보여지고 있지만) 특정의 search space 지역은 좋은 해답이 없을 가능성이 매우 높고, 또 다른 특정의 search space에는 좋은 해답이 모여있을 가능성이 높다는 것으로서, 이를 적절히 이용한 것이 MSA라고 볼 수 있다. 다만 up-hill movement의 확률치, 즉 조절변수 T를 얼마나 작게 혹은 크게 할 것인지는 많은 시행착오적 연구에 의해 얻어질 수 있다. 각 저장탱크 사용방안에 따라 이 변수를 바꾸어야 하며, 본 연구를 통해 얻은 이들 조절변수와 새로운 search방법론에 의한 결과가 매우 우수한 것으로 나타났다.

## 감 사

본 연구는 한국과학재단지정 우수 연구센터인 공정 산업의 지능자동화 연구센터의 연구비 지원에 의하여 이루어진 것으로, 연구비를 지원해 주신 연구센터에 감사드립니다.

## 사용기호

$C_{ij}$	: completion time of product i from unit 1 to unit j including transfer time $t_{ij}$
$C_{NM}$	: completion time of final product N from unit 1 to final unit M
$E_i$	: makespan as an energy function
k	: Boltzman constant

M	: number of units
N	: number of products
$P(\Delta E)$	: probability for up-hill movement
$T_i$	: temperature as a control parameter
$t_{ij}$	: processing time of product i on unit j

## 참고문헌

1. Reklaitis, G. V.: *AIChE Symp. Ser.*, **78**, 119(1982).
2. Reklaitis, G. V.: Proceedings of 4th International Symp. on PSE, Montebello, Canada(1991).
3. Ku, H. M., Rajagopalan, D. and Karimi, I.: *Chem. Engng. Prog.*, **August**, 35(1987).
4. Johnson, S. M.: *Naval Research Logistics Quarterly*, **1**, 1(1954).
5. Campbell, H. G., Dudek R. A. and Smith, M. L.: *Management Science*, **16**, B-630(1970).
6. Dannenbring, D. G.: *Management Science*, **23**, 1174(1977).
7. Wismer, D. A.: *Operations Research*, **20**, 689(1972).
8. Rajagopalan, D. and Karimi, I.: In Computer-Aided Proc. Oper., CACHE and Elsevier, New York, 679(1987).
9. Ku, H. M. and Karimi, I.: *Ind. Eng. Chem. Res.*, **30**, 163(1991).
10. Das, H., Cummings, P. T. and LeVan, M. D.: *Comput. Chem. Engng.*, **14**, 1351(1990).
11. Ku, H. M. and Karimi, I.: *Ind. Eng. Chem. Res.*, **27**, 1840(1988).
12. Metropolis, N., Rlsenbluth, A., Rosenbluth, M., Teller, A. and Teller, E.: *J. Chem. Phys.*, **21**, 1087(1953).
13. Aarts, E. and Korst, J.: "Simulated Annealing and Boltzmann Machines", John Wiley and Sons, (1989).