

Flowshop 일정계획을 위한 Simulated Annealing 알고리즘 이용 - A Study on Simulated Annealing Algorithm in Flowshop Scheduling -

우 훈 식*

Woo, Hoon Shik

임 동 순**

Yim, Dong Soon

김 철 한***

Kim, Cheol Han

Abstract

A modified simulated annealing algorithm is proposed and applied to the permutation flowshop scheduling with the makespan objective. Based on the job deletion and insertion method, a newly defined Max-min perturbation scheme is proposed to obtain a better candidate solution in the simulated annealing process. The simulation experiments are conducted to evaluate the effectiveness of the proposed algorithm against the existing heuristics and results are reported.

1. 서론

일반적으로 순열 flowshop은 주어진 n 개의 작업을 정해진 작업 순서에 의해 일련의 m 개 기계에서 처리하는 생산 시스템으로 전체 가능한 스케줄의 수는 $n!$ 이며 이로 인하여 순열 flowshop이라 불린다[1,14,16]. 순열 flowshop에서 추구하고자 하는 목표는 시스템의 특정 성능을 최적화하는 작업의 특정 순열 즉 최적 스케줄을 구하는 것이다[16].

순열 flowshop 스케줄링 문제는 지난 40여년간 스케줄링 분야에서 연구가 활발한 주제 중의 하나이다[8,16]. 잘 알려진 Johnson[5]의 2-기계 문제에 대한 makespan 최소화 알고리즘의 발표이후, 대부분의 연구는 m -기계에서의 makespan을 최소화하는 연구에 치중되었다. 하지만, 기계 수가 2보다 큰 경우 Rinnooy Kan[15]에 의해 NP complete의 특성을 갖는다고 증명되었으며, 이로 인하여 작업의 수가 많은 경우 기존의 최적화 기법은 실용적이지 못하다. 따라서, 대안으로 경험적 방법론이 Campbell *et al.*[2], Nawaz *et al.*[10]등에 의하여 제안되었으며 특히 최근에는 simulated annealing (SA)을 이용한 search 알고리즘이 활발히 연구되고 있다.

Kirkpatrick *et al.*[6]과 Cerny[3]가 combinatorial 최적화 문제 해결에 simulated annealing (SA) 방법을 도입한 이후로, flowshop 스케줄링 문제에서는 Osman & Potts[13]와 Ogbu & Smith[11,12]가 각각 다른 형태의 SA 휴리스틱을 제안하였다. Osman & Potts[13]는 Lundy &

* 시스템공학연구소 시스템통합연구부

** 한남대학교 산업공학과

*** 대전대학교 산업공학과

Mees[7]의 냉각 스케줄 관계식을 이용하여 초기 및 최종 냉각 스케줄 모수를 제안하고 기존의 알고리즘과 비교하였다. Ogbu & Smith[11]는 후보해 교체 확률이 목적함수값 변화에 독립적인 교체확률을 제안하고 해 생성 미케니즘으로는 위치 이동법을 제안하여 기존의 알고리즘과 비교 분석하였다. 또한, Ogbu & Smith[12]의 최근 결과에 의하면 Osman & Potts[13]의 결과가 Ogbu & Smith[11]의 결과보다 우월한 것으로 발표되었다.

본 연구에서는 순열 flowshop 스케줄링에서의 makespan을 최소화하기 위한 새로운 SA 방법을 제안하였다. 제안된 알고리즘은 보다 우수한 해를 얻기 위하여 작업 삭제 및 삽입 방법 기반의 후보해 생성 미케니즘을 새로 정의하였으며 기존의 연구에서 우수한 알고리즘이라고 알려진 Campbell *et al.*[2]의 CDS, Nawaz *et al.*[10]의 NEH, 그리고 Osman & Potts[13]의 SA 알고리즘과 시물레이션을 통하여 시스템 평가 수단인 makespan에 대하여 비교 분석하였다.

2. 순열 flowshop 모델링

고려된 순열 flowshop은 주어진 m 개의 기계가 $j=1, 2, \dots, m$ 으로 구성되며, 작업은 $J_i, i=1, 2, \dots, n$ 으로 구성된다[16]. 또한, 작업 J_i 의 기계 j 에서의 작업 시간은 $p(i, j)$ 로 정의되었다. 고려된 문제는 Maccathy & Liu[8]의 표기법을 따르면 $n/m/F, Perm/C_{max}$ 를 따르는 문제이다. 주어진 작업 스케줄이 J_1, J_2, \dots, J_n 이라 하고 기계 j 에서의 작업 J_i 의 시작시각을 $S(i, j)$, 종료시각을 $T(i, j)$ 라 하자. 그러면, 종료시각과 시작시각의 관계는 다음과 같다.

$$T(i, j) = S(i, j) + p(i, j) \quad (1)$$

또한, 기계 j 에서의 작업 J_i 의 최단 시작시각은

$$S(i, j) = \max\{T(i, j-1), T(i-1, j)\} \quad (2)$$

이며, 이것은 작업 J_i 는 기계 $j-1$ 에서 작업이 끝나기 전에는 기계 j 에서 작업을 시작할 수 없으며, 또한 이전 작업 J_{i-1} 이 기계 j 에서 끝나야만 하는 것을 의미한다. 즉, 작업 J_i 와 기계 j 가 모두 준비되어야 하는 것이다. 주어진 관계식 (1)과 (2)를 반복적으로 수행하여 다음의 계산 순서로 알고리즘이 진행된다.

$$T(1, 1), T(1, 2), \dots, T(1, m), T(2, 1), \dots, T(2, m), \dots, T(n, 1), \dots, T(n, m)$$

따라서 본 연구의 성능 측정 도구인 makespan은 $T(n, m)$ 이 된다.

3. Simuated annealing 알고리즘

SA 알고리즘은 열탕에서의 유리 결정의 물리적 생성공정을 컴퓨터 상에서 모의 실험하는 Metropolis *et al.*[9]의 알고리즘에서 유래한 휴리스틱 최적화 기법이다[6]. 일반적인 combinatorial 최적화 문제에 대한 시도는 Kirkpatrick *et al.*[6]과 Cerny[3]에 의해 각각 독자적으로 연구되었다. 최근에는 스케줄링 문제 등에도 폭넓게 사용되었으며, 기존의 휴리스틱 기법과 비교하여 효과적이라고 발표되었다[11,12,13].

최적화 문제에 대한 SA 알고리즘의 사용은 유리 결정을 생성하는 물리적 annealing 공정과 유사하다[6]. 물리학적 의미에서의 시스템 목적 함수는 잠재 에너지를 최소화하는 것으로 고려된 시스템은 운동 파동의 특위성 때문에 global 최소값에 도달하지 못하며 따라서 local 최소값에 머무는 것이다[9]. 이러한 상황을 극복하기 위하여, 시스템의 온도 즉 냉각 정도를 낮추면 낮출 수록 궁극적으로 시스템의 에너지를 낮출 수 있는 것이다[6]. 최적화 문제의 경우, 온도는 최적화 과정 중에 변화하는 조정 변수 C 로, 시스템의 에너지 수준은 목적함수값 $f(\cdot)$ 로 나타낼 수 있다. 이러한 물리적 annealing 공정의 냉각 스케줄 C 는 목적함수값을 증진시키기

위해 한 상태에서 다른 상태로 변환하는 search 과정과 유사하며 이러한 유사성이 시스템의 상태수가 제한적이지만 그 수가 기하급수적인 예를 들어 NP-complete의 특성을 갖는 traveling salesman 문제등에 유용하게 사용될 수 있다[3,6].

SA 방법은 기본적으로 후보해를 바꾸어 가면서 목적함수를 증진시키는 방법이다[13]. 따라서, 현재해의 주변해중 후보해를 결정하는 방법이 대단히 중요하며 이러한 방법을 해 생성 메커니즘이라 한다. 해 생성 메커니즘의 종류에는 traveling salesman 문제 해결에 사용된 2-OPT법을 변형한 상호 교환법[3]과 Ogbu & Smith가 사용한 위치 이동법[11]의 두 가지가 대표적이며, Osman & Potts[13]에 의하면 위치 이동법이 상호 교환법보다 우월한 것으로 알려져 있다.

또한, SA는 “uphill move”라는 개념을 도입하여 목적 함수를 증진시키지 못하는 해에 대하여도 교체확률 함수 가정하에 받아들인다. 이러한 uphill 해에 대해 확률적으로 인정하는 SA의 특징이 기존의 반복적 실험 방법과 구별되는 특징이며, 이러한 시도가 local 해로 부터 진행 방향을 바꾸는 수단을 제공함으로써 실제적인 문제에서의 근사 최적해를 추구하는 것이다. 따라서, 열등한 목적함수를 제공하는 후보해 Y도 현재해 X가 될 수 있도록 K번째의 후보해 생성 반복수에서 다음과 같은 교체확률 분포가 적용된다[13].

$$\Pr\{\text{replace } X \text{ by } Y\} = \min\{1, \exp(-(f(X) - f(Y)) / C_k)\} \quad (3)$$

위 관계식에서 $\exp(-(f(X) - f(Y)) / C_k)$ 은 냉각 스케줄 C_k 의 온도가 높으면 낮은 온도의 경우에 비하여 상대적으로 해가 교체될 확률이 크다. 따라서, 실험 반복수 K가 증가할 때 냉각 스케줄 C_k 의 값이 감소되도록 하면, 열등한 해가 현재해가 될 확률이 점차로 감소하게 되어 최종적으로 근사 최적해를 추구하게 된다.

4. 기존 알고리즘 고찰

순열 flowshop에서의 makespan 최소화를 위한 알고리즘은 크게 작업 순서를 차례 차례로 결정하여 스케줄을 완성하는 방법과 스케줄을 매번 바꾸어서 그중 가장 좋은 스케줄을 최종 스케줄로 결정하는 반복적 실험 방법이 있다. 본 장에서는 스케줄 완성 방법으로 잘 알려진 Campbell *et al.*[2]의 CDS 알고리즘과 Nawaz *et al.*[10]의 NEH 알고리즘을 고찰하고 스케줄 선택 방법으로 잘 알려진 Osman & Potts[13]의 SA 알고리즘을 고찰하기로 한다.

4-1. CDS 알고리즘

Campbell, Dudek & Smith[2]에 의해 제안된 CDS 알고리즘은 알고리즘 간의 성능 비교를 위한 표준으로 자주 사용되는 알고리즘이다. CDS 알고리즘은 m -기계 문제를 $(m-1)$ 개의 2-기계 문제로 분리하여 Johnson의 알고리즘을 적용하므로써 생성된 결과, 즉 $(m-1)$ 개의 스케줄을 m -기계 문제에 다시 적용하여 그 중 가장 좋은 스케줄을 선택하는 것이다. Johnson의 알고리즘을 적용하기 위한 2-기계 문제에서의 작업 시간은 다음과 같이 정의된다.

$$\text{기계 1: } \sum_{j=1}^k p(i, j)$$

$$\text{기계 2: } \sum_{j=m-k+1}^m p(i, j)$$

단, $k=1, 2, \dots, m-1$.

4-2. NEH 알고리즘

Nawaz *et al.*[10]에 의해 제안된 알고리즘은 makespan 최소화 기준에 대한 스케줄 완성

방법에 대하여 최강으로 알려져 있다. 기본적인 개념은 전체 작업 시간을 내림차순으로 정렬하여 이를 작업의 삽입 순서로 정하여 그 순서에 따라서 기존의 부분 스케줄에 삽입하여 가장 좋은 스케줄을 찾아 나가는 방법이다. 알고리즘은 다음과 같다.

단계1. 각각의 작업에 대하여 전체 작업 시간을 계산한 후 내림차순으로 정렬.

단계2. $K=2$,

정렬된 순서에서 처음 두 작업을 선택하여 최적의 결과를 생성하는 해 선택 .
(현재해로 설정).

단계3. $K=K+1$,

정렬된 순서에서 K 번째 작업을 현재해에 대하여 K 개의 slot에 삽입하여 K 개의 후보해를 생성한 후, 최적의 결과를 생성하는 해 선택 (현재해로 설정).

단계4. 만약 $K=n$ 이면, 정지.

그렇지 않으면, 단계 3으로 이동.

4-3. Osman & Potts 알고리즘

Osman & Potts[13]의 알고리즘은 현존하는 flowshop 스케줄링에서의 SA 방법 중에서 가장 우수한 결과를 제공하는 휴리스틱이다[12]. Osman & Potts[13]의 휴리스틱은 해 생성 미케니즘으로 위치교환법을 사용하였고, 냉각 스케줄 C_{K+1} 은 Lundy & Mees[7]의 다음 스케줄을 사용하였다.

$$C_{K+1} = C_K / (1 + \beta C_K) \quad (4)$$

단, $K=1, 2, \dots, N-1$.

관계식 (4)에서 β 는 상수이며 Connolly[6]에 의하여 다음과 같이 정의된다.

$$\beta = (C_1 - C_N) / (C_1 C_N (N-1)) \quad (5)$$

또한, Osman & Potts[13]는 다음과 같이 초기 냉각 스케줄 C_1 및 최종 냉각 스케줄 C_N 을 제안하였다.

$$C_1 = \sum_{i=1}^n \sum_{j=1}^m P(i, j) / 5mn$$

$$C_N = 1 \quad (6)$$

다음은 Osman & Potts[13]의 알고리즘을 단계별로 정리한 것이다.

단계 1. 반복수 $K = 1$,

초기해 X 를 임의로 선정한다. 초기해 X 를 현재해로 정한다.

단계 2. $K = K + 1$,

해 생성 미케니즘에 따라서 후보해 Y 를 생성한다.

단계 3. 교체 확률 분포 (3)에 따라 현재해 X 를 후보해 Y 로 교체한다.

단계 4. 만약 $K \geq N$ 이면 정지하고, 그렇지 않으면 단계 2로 이동한다. 단, N 은 계획된 전체 반복수.

5. 제안 알고리즘

본 제안 알고리즘의 기본적인 개념은 기존 알고리즘의 해 생성 미케니즘을 개선하는 것이다. 기존의 SA 알고리즘에서 사용된 해 생성 미케니즘의 종류에는 이미 기술한 바와 같이 상호 교환법과 위치 이동법이 있다[11]. 상호 교환법은 해 스케줄의 위치값 2개를 난수로 선택하여

상호 교환하는 방법이고, 위치 이동법은 난수로 선택된 두 위치에 대하여 한 위치에서 다른 위치로 이동시키는 방법이다. 예를 들어, 주어진 현재해가 (1,2,3,4,5)이고 난수로 결정된 위치 a, b , ($a, b \leq 5, a \neq b$)의 값이 ($a=1, b=4$)라 하자. 그러면, 상호 교환법에 의한 후보해는 (4,2,3,1,5)가 되고 위치이동법에 의한 후보해는 (2,3,4,1,5)가 된다. 이와같이 기존 알고리즘에서 사용된 해 생성 메커니즘은 현재해의 주변해 중에서 임의의 난수를 생성하여 후보해를 결정하므로, 주변해 모두에게 균등한 확률 기회를 주는 방법이다.

본 연구에서는 이러한 균등 확률 할당 방법을 탈피하여 현재해의 주변해 중에서 후보해를 직접 결정하는 방법을 제안하였다. 제안된 후보해 생성 메커니즘은 현재해에서 특정 작업을 제외하였을 때 makespan의 감소치가 가장 큰 작업을 선정하고 다시 선정된 작업을 선정된 작업이 제외된 스케줄의 모든 삽입 위치에 삽입하여 makespan의 증가치가 가장 작은 위치를 선정하는 방법으로 Max-min 알고리즘이라 부를 수 있다. 제안된 Max-min 알고리즘의 장점은 SA 알고리즘의 진행 방향을 현재해의 주변해 중에서 우수한 결과를 나타내는 해로 직접 진행시키므로 기존의 균등 확률 할당 방법보다 우수한 결과의 후보해를 도출할 수 있다는 것이다. 제안 알고리즘에서의 냉각 스케줄은 Osman & Potts[13]에서 사용된 관계식 (4), (5), (6)을 이용하였으며, 다음과 같이 제안 알고리즘을 단계별로 정리하였다.

[제안 알고리즘]

- 단계 1. 임의로 최초해 X 를 선정하고 $f(X)$ 를 산정한다. $K=1$ 로 정한다.
- 단계 2. 아래에 정의된 Max-min 알고리즘에 따라서, 현재해인 X 의 주변해로부터 후보해 Y 를 생성한다. 목적함수 $f(Y)$ 를 산정하고 K 를 $K=K+2n$ 으로 정한다.
- 단계 3. 만약 $K \geq N$ 이면 정지하고, 그렇지 않으면 단계 4로 이동한다.
- 단계 4. 만약 $f(X) > f(Y)$ 이면 Y 를 현재해로 정하고 단계 5로 이동한다. 그렇지 않으면 후보해 Y 를 $\exp(-(f(X)-f(Y))/C_k)$ 의 확률로 현재해로 정의한 뒤 단계 5로 이동한다.
- 단계 5. 만약 단계 4에서 Y 가 현재해로 정의되면 단계 2로 이동, 단계 4에서 Y 가 현재해로 정의되지 않으면 임의로 새로운 후보해 Y 를 생성하고, K 를 $K=K+1$ 로 정한 후, 단계 3으로 이동.

[Max-min 알고리즘]

- 단계 1. 현재해 X 의 작업중에서 다음의 식을 만족하는 작업 a , $1 \leq a \leq n$ 를 선택한다.

$$\text{Max}_a f(X) - f(X_{-a})$$

단, X_{-a} 는 현재해 X 에서 작업 a 를 제외한 스케줄.

- 단계 2. 다음의 식을 만족하는, 단계 1의 작업 a 가 삽입될 스케줄 X_{-a} 의 위치 b 를 결정한다.

$$\text{Min}_b f(X_{a,b}) - f(X_{-a})$$

단, $X_{a,b}$ 는 스케줄 X_{-a} 의 위치 b , $b \in \{1, 2, \dots, n+1\}$, $b \neq a$ 에 작업 a 가 삽입된 새로운 스케줄.

6. 시뮬레이션을 이용한 실험

본 연구에서는 CRAY C90 슈퍼 컴퓨터를 이용하여 Fortran으로 코딩하여 제안 알고리즘을 기존의 알고리즘인 CDS, NEH, 그리고 Osman & Potts의 알고리즘과 비교하였다. 본 실험

에서 고려된 작업의 수는 10, 15, 20, 25, 30이며, 기계의 수는 10, 15, 20, 25이고 또한 작업 시간은 uniform 분포 (1,99)를 사용하였다. 각 작업수 및 기계수별로 100회씩 실험하였으며, 따라서 고려된 총 문제수는 8000이다. 또한, CDS와 NEH 알고리즘은 스케줄 완성 방법이므로 반복수가 필요없고, Osman & Potts와 제안 알고리즘은 스케줄 선택 방법이므로 반복수를 $N=50,000$ 으로 정하고 실험하였다.

알고리즘의 성능을 평가하기 위하여, 알고리즘의 결과중에서 가장 우수한 결과를 기준으로 하였다. 따라서, 최우수 결과로부터의 평균 편차와 최우수 결과를 나타낸 빈도수를 성능 평가 기준으로 정하였다. 최우수 결과로부터의 상대 평균 편차는 다음과 같이 정의된다:

$$\sum_{i=1}^{100} [(\text{알고리즘결과}_i - \text{최우수결과}_i) / \text{최우수결과}_i] / 100$$

알고리즘간의 성능을 비교한 시뮬레이션 결과가 표1과 표2에 정리되었다. 표1은 작업수별, 기계수별로 CDS, NEH, Osman & Potts, 그리고 제안 알고리즘을 비교한 시뮬레이션 결과이다. 비교된 알고리즘의 상대평균편차는 전체평균값으로 CDS 알고리즘 8.42%, NEH 알고리즘 2.72%, Osman & Potts 알고리즘 0.47%, 그리고 제안 알고리즘 0.33%를 나타냈으며 최우수해를 나타낸 빈도수는 100회의 반복실험중 평균으로 CDS 알고리즘 0.2회, NEH 알고리즘 3.2회, Osman & Potts 알고리즘 51.1회, 그리고 제안 알고리즘 61회를 기록하였다. 또한, 작업수 및 기계수별로 구성된 총 20개의 문제 set 중에서 상대평균편차 기준에 대해서는 스케줄 완성 방법인 CDS와 NEH 알고리즘은 0회, Osman & Potts 알고리즘은 8회, 그리고 제안 알고리즘은

표 1. 최우수해 기준의 알고리즘 성능 평가 비교

작업수	기계수	CDS		NEH		Osman&Potts		제안알고리즘	
		평균편차 (%)	최우수해 빈도수	평균편차 (%)	최우수해 빈도수	평균편차 (%)	최우수해 빈도수	평균편차 (%)	최우수해 빈도수
10	10	6.31	2	2.65	7	0.10	89	0.25	74
	15	5.34	0	2.01	11	0.17	80	0.24	68
	20	5.09	2	2.32	4	0.18	76	0.22	68
	25	4.15	0	1.66	4	0.10	79	0.31	58
15	10	9.02	0	3.08	5	0.35	60	0.43	55
	15	8.70	0	2.96	5	0.43	47	0.39	56
	20	7.12	0	2.82	2	0.41	50	0.38	57
	25	6.56	0	2.68	5	0.49	41	0.24	62
20	10	11.15	0	3.41	1	0.43	55	0.45	52
	15	9.35	0	2.88	1	0.51	44	0.39	57
	20	8.25	0	2.77	3	0.67	31	0.18	68
	25	7.91	0	2.60	2	0.69	33	0.19	67
25	10	10.71	0	3.02	2	0.50	57	0.53	52
	15	9.83	0	3.11	1	0.73	35	0.25	67
	20	9.56	0	2.94	2	0.70	33	0.24	67
	25	8.89	0	2.68	2	0.75	34	0.32	65
30	10	10.78	0	2.78	1	0.38	60	0.69	45
	15	10.40	0	2.79	3	0.60	46	0.44	53
	20	9.93	0	2.72	0	0.62	41	0.33	60
	25	9.28	0	2.52	3	0.66	31	0.22	68

표 2. 반복수에 따른 알고리즘 성능 평가 비교

반복수	Osman & Potts			제안 알고리즘		
	평균편차 (%)	최우수해 빈도수	평균 CPU시간	평균편차 (%)	최우수해 빈도수	평균 CPU시간
10000	0.4765	56.00	0.2160	0.6362	52.60	0.2158
20000	0.4810	55.25	0.4311	0.5181	54.75	0.4313
30000	0.4574	54.55	0.6469	0.4071	58.25	0.6477
40000	0.4578	52.50	0.8617	0.3605	61.00	0.8627
50000	0.4633	52.15	1.0777	0.3255	62.25	1.0786

12회가 타 알고리즘에 비하여 우수하였으며, 최우수해 빈도수 기준에 대해서도 CDS와 NEH 알고리즘은 0회, Osman & Potts 알고리즘은 8회, 그리고 제안 알고리즘은 12회가 우수하였다. 따라서, 알고리즘의 우수 정도는 제안, Osman & Potts, NEH, CDS 알고리즘으로 순서화할 수 있다. 표1에서 특이한 점은 작업수가 10인 경우와 같이 소규모 문제에서는 Osman & Potts의 알고리즘이 보다 우수한 결과를 생성하였다. 따라서, 문제의 형태에 따라 제안 알고리즘과 Osman & Potts 알고리즘을 적용할 수 있을 것이다.

표2는 표1의 알고리즘 비교실험에서 우수한 결과를 생성한 SA 알고리즘간의 반복수에 따른 알고리즘 특성을 평가한 것이다. 즉, SA 알고리즘인 Osman & Potts 알고리즘과 제안 알고리즘에 대하여 반복수를 $N=10000, 20000, \dots, 50000$ 으로 정하고 표1의 문제 Set에 대하여 두 알고리즘을 비교하여 상대평균편차, 최우수해빈도수, 그리고 소요 CPU 시간을 기록하였다. 비교 실험 결과에서 주목할 만한 점은 반복수가 증가하면 할수록 제안 알고리즘의 우수성이 뚜렷하다는 것이다. 특히 반복수가 $N=30000$ 이상인 경우 제안 알고리즘이 Osman & Potts 알고리즘보다 우수한 결과를 생성하였다. 또한 소요 CPU 시간은 각 반복수에 따른 알고리즘의 평균소요시간을 기록한 것으로, 평균으로 단일 문제에 소요된 제안 알고리즘의 소요시간은 0.6472초로 Osman & Potts 알고리즘의 0.6467초와 별 차이가 없었다.

7. 결론

본 연구에서는 n -작업 및 m -기계로 구성된 순열 flowshop 문제에서 makespan을 최소화 하기 위한 새로운 simulated annealing 방법을 제안하였다. 제안된 알고리즘에서는 보다 정확한 해를 얻기 위하여 작업 삭제 및 삽입 방법 기반의 후보해 생성 미케니즘인 Max-min 알고리즘을 새로 정의하였으며 기존 알고리즘과의 비교를 시뮬레이션을 이용하여 광범위하게 수행하였다. 비교 실험 결과는 제안 알고리즘의 기존 알고리즘에 대한 우수성을 보여주었으며 특히 실험 반복수가 증가하는 경우 타알고리즘에 비하여 제안 알고리즘의 우수성이 뚜렷하였다.

참고문헌

- [1] Baker, K. R., *Introduction to sequencing and scheduling*, Wiley, New York, 1974.
- [2] Campbell, H., R. Dudek, and M. Smith, "A heuristic algorithm for the n-job, m-machine sequencing problem", *Management Science*, Vol. B 16 (1970), pp. 630-637.

† 본 연구를 위해 슈퍼컴퓨팅자원을 지원한 시스템공학연구소 슈퍼컴퓨터센터에 감사드립니다.

- [3] Cerny, V., "A thermodynamical approach to the travelling salesman problem: an efficient simulation algorithm", *Journal of Optimization Theory and Application*, Vol. 45 (1985), pp. 41-51.
- [4] Connolly, D., "An improved annealing scheme for the QAP", *European Journal of Operational Research*, Vol. 46 (1990), pp. 93-100.
- [5] Johnson, S., "Optimal two- and three-stage production schedules with set up times included", *Naval Research Logistics Quarterly*, Vol. 1 (1954), pp. 61-68.
- [6] Kirkpatrick, S., C. D. Gelatt Jr., and M. P. Vecchi, "Optimization by simulated annealing", *Science*, Vol. 220 (1980), pp. 671-680.
- [7] Lundy M. and A. Mees, "Convergence of an annealing algorithm", *Mathematical Programming*, Vol. 34 (1986), pp. 111-124.
- [8] Maccarthy, B. and J. Liu, "Addressing the gap in scheduling research: a review of optimization and heuristic methods in production scheduling", *International Journal of Production Research*, Vol. 31 (1993), pp. 59-79.
- [9] Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, and A. H. Teller, "Equation of state calculation very fast computing machines", *Journal of Chemical Physics*, Vol. 21 (1953), pp. 1087-1092.
- [10] Nawaz, M., E. Enscore, and I. Ham, "A heuristic algorithm for the m-machine, n-job flowshop sequencing problem", *OMEGA*, Vol. 11 (1983), pp. 91-95.
- [11] Ogbu, F. A., and D. K. Smith, "The application of the simulated annealing algorithm to the solution of the n/m/Cmax flowshop problem", *Computers and Operations Research*, Vol. 17 (1990), pp. 243-253.
- [12] Ogbu, F. A., and D. K. Smith, "Simulated annealing for the permutation flowshop problem", *OMEGA*, Vol. 19 (1990), pp. 64-67.
- [13] Osman, I., and C. Potts, "Simulated annealing for permutation flow-shop scheduling", *OMEGA*, Vol. 19 (1989), pp. 551-557.
- [14] Pinedo, M., *Scheduling: Theory, algorithms, and systems*, Prentice Hall, Englewood Cliffs, New Jersey, 1995.
- [15] Rinnooy Kan, A., *Machine scheduling problems*, Nijhoff, The Hague, 1976.
- [16] Woo, Hoon-Shik, and Dong-Soon Yim, "A heuristic algorithm for mean flowtime objective in flowshop scheduling", *Computers and Operations Research*, to be appeared.