동일하지 않는 병렬기계 시스템에서 지연작업수를 최소화하는 Tabu Search 방법

전태웅* · 강맹규**

Tabu Search Methods to Minimize the Number of Tardy Jobs in Nonidentical Parallel Machine Scheduling Problem

Tae-Woong Chun* · Maing-Kyu Kang**

ABSTRACT

This paper presents a Tabu Search method to minimize a number of tardy jobs in the nonidentical parallel machine scheduling. The Tabu Search method employs a restricted neighborhood for the reduction of computaion time. In this paper, we use two different types of method for a single machine scheduling. One is Moore's algorithm and the other is insertion method. We discuss computational experiments on more than 1000 test problems.

1. 서론

본 연구는 n개의 작업과 m대의 기계가 있는 동일하지 않는 병렬기계 시스템에서 일정계획의 최적화 문제를 다룬다. 동일하지 않는 병렬기계 (nonidentical parallel machine) 시스템이란 각 기계의 성능이 다르거나 또는 노후화의 정도에 따라 i번째 ($i=1,\cdots,n_i$) 기계에 배정된 j 번째 ($j=1,\cdots,n_i$)작업 J_{ij}의 가공시간 P_{ij}가 각 기계마다 다르고, 작업들이 병렬 처리되는 시스템이다(여기서 n_i는 i번째 기계에 배정된 작업의 수 이다.). 이 시스템의 예로는 제조시스템의 병목공정에서 작업처리와 병원시스템의 환자처리 등이다[4].

본 연구에서 가정한 사항은 작업분할은 허용하지 않으며, 시간 0에서 모든 기계가 이용가능하고 각 작업의 가공시간과 납기시간 D_{ij}는 주어진다. 일정계획에 대한 수행도는 지연작업수(number of tardy jobs, N_T)를 최소화하는 것으로 지연작

* 조선대학교 산업공학과
** 한양대학교 산업공학과

업(tardy job) T_{ij}는 작업의 완성시간 C_{ij}가 납기 D_{ij}를 초과할 때 발생된다. 따라서 본 연구의 목적식은 식 (1)로 표현된다.

$$\text{최소화 } N_T = \sum_{i=1}^{m}\sum_{j=1}^{n_i} T_{ij} \qquad (1)$$

$$T_{ij} = \begin{cases} 1, & \text{if } C_{ij} > D_{ij} (j=1,\cdots,n_i),\ \forall i \\ 0. & \end{cases}$$

지연작업수를 최소화하는 문제를 다룬 연구로 Moore[11]는 단일기계 일정계획 문제에서 최적해를 제공하는 방법으로 Moore의 방법(일명 Hodgson의 방법)을 개발하였으며, Villarreal과 Bulfin[13]은 단일기계 일정계획 문제에서 가중 지연작업수를 최소화하는 분지한계법을 개발하였다. Lawler and Martel[10]은 두개의 균일한 병렬기계(uniform parallal machine) 일정계획 문제에서 동적계획법을 사용하였다. Ho와 Chang [8]은 분할 작업이 허용되지 않는 동일한 병렬기계 일정계획 문제에서 효율적인 휴리스틱 방법을 개발하였다.

본 연구에서 다룰 동일하지 않는 병렬기계 일정계획 문제에 대한 연구는 아직까지 문헌에 발표되지 않고 있다. 병렬기계 일정계획 문제의 최적해를 구할 때 필요한 계산량은 n개의 작업을 m대의 기계에 배정할 때의 계산량과 각 기계에서의 일정계획의 최적화에 필요한 계산량의 합이다. N_T 문제는 단일기계 일정계획 문제에서 최적화 방법이 존재하므로 이 계산량을 제외하더라도 n개의 작업을 m대의 기계에 배정할 때 필요한 계산량은 $o(m^n)$이며 NP-complete 문제이다[8].

최근 복잡한 조합최적화 문제를 해결하기 위해 개발된 Tabu Search(TS) 방법을 일정계획 문제에 적용하여 좋은 해를 구하고 있다. 일정계획에 적용한 연구로 Laguna 등[9]의 단일기계 일정계획 문제, Barnes와 Laguna[3]의 복수기계 일정계획 문제, Taillard[12]의 흐름생산 일정계획 문제, Dell'Amico와 Trubian[6] 및 Barnes와 Chambers[4]의 개별생산 일정계획 문제와 전과 강[1]의 병렬기계 일정계획 문제 등이 있다.

본 연구의 목적은 동일하지 않는 병렬기계 시스템에서 지연작업수의 최소화 문제에 TS 방법을 적용하여 좋은 해를 찾고자 한다. 2장에서는 본 연구와 관련된 TS 방법을 간단히 소개한다. 3장과 4장에서는 본 연구에서 적용한 TS 방법의 설계 및 제한된 이웃해집단의 구성방법을 보이고 5장에서는 실험결과를 요약한다.

2. 일반적인 TS 방법

TS 방법은 이웃해 탐색방법(neighbour search method)과 탐색절차가 유사하므로 먼저 이웃해 탐색방법의 탐색절차를 간단히 설명한다. 다음과 같은 최적화 문제가 있다고 하자.

$$\begin{aligned} &\text{최소화} && F(s) \\ &\text{제약조건} && s \in X \end{aligned} \qquad (2)$$

여기서 X는 해의 공간이며, F(s)는 목적함수이다. 초기해 $s' \in X$를 현재해 s로하여 탐색을 시작한다. 탐색의 매 단계마다 이 s를 이동(move)하여 이웃해집단 N(s)를 생성하고, 이웃해 $s_i \in N(s)$중에서 $F(s_i)$가 최소가 되는 s'를 선택한다. 이때 만약 $F(s') < F(s)$이면 s'를 현재해로 대체하여 탐색을 계속하고, 그렇지 않으면 탐색을 끝낸다. 이웃해 탐색방법에서 찾아진 해는 최적해가 보장되지 않는다.

TS 방법의 기본적인 탐색절차는 초기해 $s_0 \in X$

를 현재해로 탐색을 시작하여 이웃해집단 N(s)를 생성하고 그 중에서 F(s)가 최소가 되는 s'를 선택한다. 이때 F(s')<F(s)이면 s'를 현재해로 하여 탐색을 진행하는 절차는 이웃해 탐색 방법과 같다. 그러나 TS 방법은 선택한 s'가 F(s')>F(s) 일지라도 탐색공간을 넓히기 위하여 s'를 현재해로 하여 탐색을 계속한다. 그리고 해의 반복이나 사이클링을 방지하기 위해 s'를 일정기간 동안 기억장소(tabu list) T에 저장하여 새로 찾아진 s'가 T안의 해(이 해를 tabu라고 함)가 되지 않도록 한다. 또한 더 나은 해를 찾기 위하여 새로 찾아진 s'가 T안의 해일지라도 s'를 선택할 수 있는 기준(이를 희망기준(aspiration criteria)이라고 함)을 사용하여 s'가 이 기준을 만족하면 이를 선택한다. TS 방법은 이상과 같은 탐색절차를 일정 횟수동안 반복하여 해를 찾는 방법이다.

3. 적용한 TS 방법

3. 1 이동방법

그림 1은 본 연구에서 적용한 TS의 이동방법을 설명하기 위한 예로서, TS 방법을 m=3, n=10인 지연작업수의 최소화 문제에 적용하여 해를 탐색할 때 찾아진 하나의 현재해 s를 나타낸다.

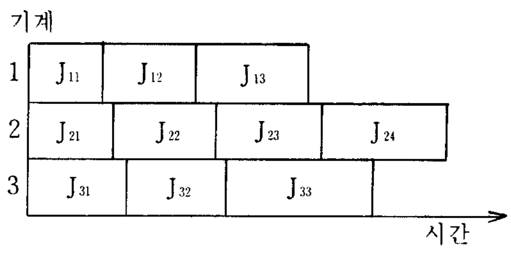

그림 1. m=3, n=10일 때 현재해

본 연구는 그림 1과 같은 현재해 s에서 이웃해 s_i를 생성하는 이동방법으로, 다른 두 기계에 있는 작업을 교환시키는 교환이동(swap move)과 한 기계에 있는 하나의 작업을 다른 기계로 이동시키는 삽입이동(insert move)을 사용한다. 따라서 s를 교환이동하면 각 기계에 배정되어 있는 작업수가 s와 s'에서 같으나 삽입이동은 s'에서 달라진다.

예를들어, s에서 임의로 J_{11}을 택하여 교환이동하면 교환되는 작업들의 쌍은 (J_{11}, j_{21}), (J_{11}, j_{22}), (J_{11}, j_{23}), (J_{11}, j_{24}), (J_{11}, j_{31}), (J_{11}, j_{32}) 및 (J_{11}, j_{33})가 되며 이 경우 생성되는 이웃해집단의 크기는 66개가 된다. 이를 일반화하면 n개의 작업에 대한 이웃해집단의 크기는 $n^2-\sum_{i=1}^{m} n_i^2$ 이다. 또한 s에서 임의로 J_{11}을 택하여 삽입이동하면 m=2, m=3의 기계에 배정된 작업수가 하나 증가하며 m=1에서의 작업수는 하나 감소한다. 따라서 J_{11}을 삽입이동하여 생성할 수 있는 이웃해는 m−1인 두 개가 되며, n개의 작업을 삽입이동하면 (m−1)개의 이웃해가 생선된다.

위와 같은 본 연구의 이동방법은 각 기계에 배정될 작업이 최적화되도록 함으로써 최적해가 찾아지도록 설계하였다. 즉 단일기계 일정계획 문제에서 최적해를 제공하는 방법이 존재하므로 각 기계에 배정될 작업이 최적화되면 최적해가 구해질 수 있다. 그러나 본 연구의 이동방법은 생성되는 이웃해가 n과 m이 커짐에 따라 매우 커지므로 계산시간을 절약하기 위하여 본 연구에서는 제한된 이웃해집단을 사용한다.

3. 2 단일기계 일정계획 방법

그림 2는 현재해를 교환이동할 때 기계 x의 J_{xi}과 기계 y의 J_{yj}가 교환 대상이 된 현재해를 나타낸다.

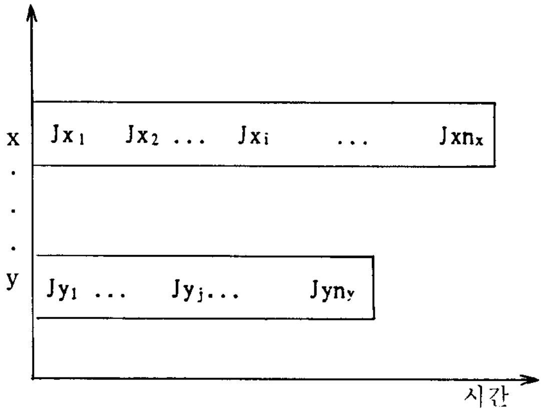

그림 2. J_{xi}와 J_{yj}의 교환 전 현재해

이 두 작업을 교환하면 J_{xi}는 기계 y의 J_{yj} 위치, J_{yj}는 기계 x의 J_{xi}의 위치로 와서 하나의 이웃해를 생성한다. 이때 두 작업은 N_T의 최소화가 보장된 교환이 아니고 단순히 새로운 해를 얻기 위한 것이므로 N_T가 최소화되도록 각각의 기계에 대한 작업순서를 다시 정해야 한다. 따라서, 현재해를 교환이동하여 작업들이 교환되더라도 교환된 기계에 대해서 작업순서를 최적화하는 단일기계 일정계획 방법을 적용해야 한다.

본 연구에서는 단일기계 일정방법으로 최적해를 제공하는 Moore 방법을 사용한 방법(이를 TS_1이라고 함)과 최적해를 보장하지 못하지만 계산시간을 절약하기 위한 방법으로 끼워넣기 방법을 사용한 방법(이를 TS_2라고 함)을 적용하고 실험한다.

Moore 방법을 간단히 설명하면 다음과 같다. n개의 작업을 두 개의 집합으로 분할한다. 첫번째 집합은 지연이 발생하지 않는 작업들로 구성하고 이 집합내의 작업순서는 최소납기 우선순위로 정렬한다. 두번째 집합은 지연이 발생되는 작업들로 구성하고 임의로 작업순서를 정한다. 이렇게 구성한 작업순서는 지연작업수를 최소화시키며 이 방법의 계산량은 $O(n \log n)$이다[2, 11].

본 연구에서 사용하는 끼워넣기 방법은 하나의 해에서 다른 해를 생성하는 방법으로 많이 사용하는 삽입방법과 같다. 즉, 그림 2에서 J_{yj}를 기계 x에 나열되어 있는 작업들(J_{x1}, J_{x2}, J_{x3},…,J_{xn_x}) 앞에 삽입시키는 방법을 사용하여 기계 x에 발생가능한 n_x개의 해를 얻고 n_x개의 해 중 N_T를 최소화하는 해를 찾아 그 기계에서의 작업순서로 한다. 이 방법의 계산량은 $O(n)$이다.

3. 3 적용된 TS 방법의 구성요소

본 연구의 TS_1, TS_2 방법에서는 Tabu 리스트내에 이동요소인 작업번호를 저장한다. Tabu 제한으로는 교환에 참여한 작업은 Tabu 리스트의 크기인 Tabu tenure |T|동안 이동할 수 없도록 하였으며 |T|는 일반적으로 많이 사용하는 7로 하였다. 희망기준으로 새로 찾아진 해의 값이 최선해보다 좋으면 이해를 현재해로 선택한다.

일반적으로 발견적 기법은 초기해에 영향을 받기 때문에 본 연구는 좋은 해를 구하기 위해 다음과 같은 방법으로 초기해를 구한다. 이 방법을 본 연구에서는 LIST라고 한다.

단계1. n개의 작업을 최소 납기의 오름차순으로 순서화 한다.

단계2. 순서화된 작업을 차례대로 현재의 작업부하가 가장 작은 기계에 할당한다. 만약 n개의 작업이 모두 배정되면 단계 3으로 간다. 그렇지 않으면 단계 2로 간다.

단계3. i번째 기계(i=1,…, m)에 대해 Moore 방법을 적용한다.

종료조건으로는 최대 반복 횟수 max_iter를 n으로 한다.

본 연구에서 제안하는 TS 방법은 그림 3과 같다.

```
Natations : X : set of feasible solution.
            f( . ) : N_T
            niter : n.
            nb_iter : the number of current iteration.
            best_sol : the best solution in current search process.
  Problem : Find x(x∈X) such that f(x) is minimum.
  Algorithm :
     initialization :
            -generate the initial solution s(s∈X) by LIST ;
            -bset_sol : =f(s) ;
            -nb_iter : =0 ;
            -T : ={φ}
     while (nb_iter<max_iter)
            -generate neighbours s_i of current solution s with
              move (s->s_i) ∉ T or with f(s_i)<best_sol ;
            -f(s_i) is caculated by insertion method for one machine
                   scheduling ;
                   or by Moore's algorithm ;
            -let s' be the best neighbour generated ;
            -update tabu list ;
            -if f(s')<best_sol then
                 -best_sol : =f(s') ;
                 -s : =s' ;
            -nb_iter : =nb_iter+1 ;
     end while
```

그림 3. 적용한 TS 방법

4. 제한된 이웃해의 구성

본 연구의 삽입이동과 교환이동은 현재해로부터 생성가능한 이웃해집단의 크기가 매우 크다. 그러므로 본 연구는 현재해에서 교환이동을 할 경우 생성가능한 이웃해집단의 크기를 줄이는 방법으로 제한된 이웃해집단을 구성하여 사용한다. 즉, 본 연구는 교환될 기계 x의 작업과 교환하려는 기계 y의 작업순서를 고려하여 교환될 작업을 하나만 선택한다. 그림 4는 본 연구에서 구성한 제한된 이웃해 집단의 구성 형태이다. 그림 4에서 J_{xi}의 교환 대상의 작업은 기계 y(y=1,…, m, y≠x)의 J_{yj}가 된다.

5. 실험결과 및 분석

본 연구에서 적용한 TS 방법은 단일기계 일정계획을 위해 Moore 방법을 적용한 TS_1 방법과 끼워넣기 방법을 적용한 TS_2이다. TS_1 방법과 TS_2 방법의 효율을 평가하기 위해 다음과

기계 번호	작 업 순 서		
	1 . . . j−2 j−1	J	J+1 J+2 . . . n_i
⋮		⋮	
x	J_{x1} . . . J_{xj-2} J_{xj-1}	J_{xj}	J_{xj+1} J_{xj+2} . . .
⋮		⋮	
y	J_{y1} . . . J_{yj-2} J_{yj-1}	J_{yj}	J_{yj+1} J_{yj+2} . . .
⋮		⋮	
m	J_{m1} . . . J_{mj-2} J_{mj-1}	J_{mj}	J_{mj+1} J_{mj+2} . . .

그림 4. 제한된 이웃해집단의 구성

표 1. 최적해와 비교

방법	n=10		n=15		n=20	
	N_T	오차	N_T	오차	N_T	오차
TS_1	2.46	0.016	3.35	0.031	3.47	0.048
TS_2	2.48	0.024	3.38	0.040	3.52	0.063
OPTIMAL	2.42	0.000	3.25	0.000	3.31	0.000

같이 실험을 하였다. 실험에 사용한 문제는 가공시간 P_{ij}=Uniform[1, 100], 납기시간 D_{ij}=Uniform[1, $2n\bar{P}/mq$)에 의하여 발생시켰다. 여기서 q는 납기의 완급정도를 나타내는 인자로 q=1,⋯, 5로 나타내며 q값이 적으면 납기가 느슨하고 q가 크면 납기가 급한 것을 나타낸다[7].

본 연구는 m과 n의 변화에 따른 TS_1, TS_2 방법의 효율을 평가하기 위하여 기계대수 m=2, 3, 5로 하고, 작업의 수는 n/m의 비율이 5, 10, 15, 20가 되도록 하였다. 또한 납기의 완급정도가 발생되는 문제에 균등히 포함되도록 q=2, 3.5, 5로 하여 각각의 문제에 대해서 30회 반복하였다. 본 문제에 TS_1, TS_2 방법을 적용할 경우, 구해지는 해의 값과 최적해와의 차이를 평가하기 위해 m=2일 때, n=10, 15, 20인 문제를 만들고 각각의 문제에 대하여 100회 반복하였다. 본 연구에서는 최적해를 열거법에 의해 구하였고, 사용한 컴퓨터는 486DX(50MHz)이며 사용된 컴퓨터 언어는 FORTRAN-77이다.

표 1은 TS_1 방법, TS_2 방법에서 구한 해와 최적해와의 차이를 (TS 방법의 해의 값/최적해의 값−1)*100으로 구한 오차로 나타낸 것이다. 표 1에서 나타난 바와 같이 TS_1과 TS_2는 최적해와의 차이가 n이 커짐에 따라 약간씩 증가한다. 그러나 계산시간면에서 m=2, n=20인 경우 최적해를 구하기 위해 소비된 시간은 약 27분이 소요되었으나 TS_1, TS_2 방법 모두 1초가 넘지 않았다. 또한 본 연구의 실험에서 300 문제 중 TS_1 방법은 22 문제, TS_2 방법은 29 문제에 대해서 최적해와 다르게 나타났다. 따라서

표 2. 실험결과

m	m	N_T		계산시간(초)	
		TS_1	TS_2	TS_1	TS_2
2	10	2.46	2.48	0.17	0.16
	20	3.47	3.52	0.66	0.33
	30	7.44	9.05	0.88	0.74
	40	9.50	10.15	1.43	1.24
3	15	3.10	3.38	0.22	0.20
	30	5.27	5.40	1.60	1.15
	45	7.80	7.93	4.61	3.46
	60	15.60	16.50	13.79	9.55
5	25	6.66	6.84	0.51	0.47
	50	9.14	9.80	7.03	5.01
	75	16.90	17.19	28.01	20.98
	100	21.37	22.30	89.31	63.20

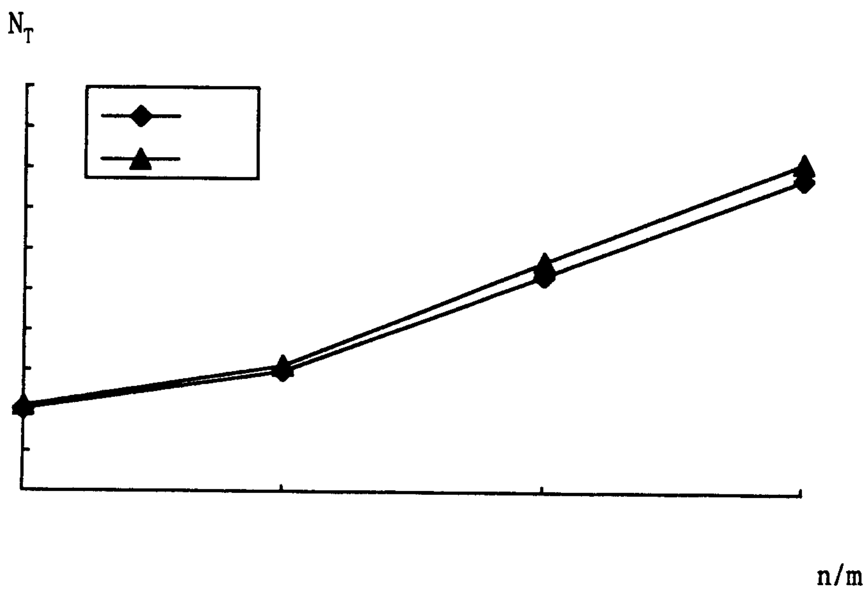

그림 5. n/ m의 변화

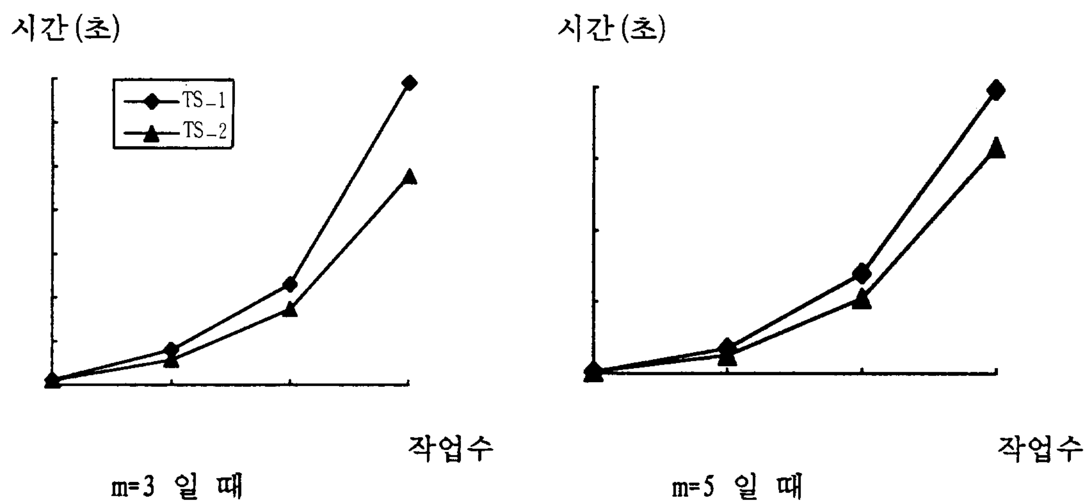

그림 6. TS_1과 TS_2의 계산시간

TS_1과 TS_2 방법은 적은 문제에 대해서는 최적해와 근사함을 알 수 있다. 표 2는 본 연구의 실험결과를 나타낸다.

그림 5는 한 기계에 평균적으로 작업이 n/m 개가 배정될 경우 TS_1 방법과 TS_2 방법을 비교한 것으로, n/m의 값이 커질수록 TS_1 방법이 TS_2 방법보다 평균값이 낮다. 그러나 그 차이는 n/m이 20일 경우 TS_1의 해의 값에 0.052로 매우 적은 것으로 나타난다(차이=TS_2 TS_1)/TS_1).

그림 6은 최대 반복횟수를 n으로 할 때 TS_1 방법과 TS_2 방법의 계산시간을 비교한 것으로, 계산시간 면에서는 TS_2가 n/m 및 문제의 크기가 클수록 계산시간이 적다. 따라서 문제의 크기가 클 수록 계산시간 면에서 TS_2 방법이 효율적이므로 기계대수와 작업의 수가 많을 경우 본 연구의 TS 방법을 적용할 때 TS_2 방법을 사용하는 것이 바람직하다.

5. 결 론

동일하지 않는 병렬기계 시스템에서 지연작업수를 최소화하는 문제는 NP-complete 문제이다. 본 연구는 이 문제의 좋은 해를 얻기 위하여 TS 방법을 적용하였다. 본 연구의 TS 방법은 이동방법으로 교환이동과 삽입이동을 사용하였으며, 계산시간을 줄이고자 제한된 이웃해집단을 구성하여 사용하였다. 또한 단일기계에서 최적의 작업순서를 결정하기 위해 Moore 방법을 사용하는 TS_1 방법과 끼워넣기 방법을 사용한 TS_2 방법을 개발하고 이 두 방법을 비교하는 실험을 하였다. 초기해로 본 문제에 적합한 최소부하 할당법과 Moore 방법을 사용한 LIST 방법을 TS_1, TS_2 방법에 적용하였다.

본 연구의 실험에서는 적용된 TS_1, TS_2 방법은 n이 커질수록 최적해와 약간의 차이가 발생되지만 적은 문제에서는 최적해와 근접하였다.

TS_1 방법과 TS_2 방법을 비교할 때 n/m이 커질수록 TS_1 방법의 해가 약간 좋게 나타나지만 그 차이는 매우 적게 나타남을 알 수 있었다. 계산시간 면에서는 TS_1 방법보다 TS_2 방법이 문제의 크기가 클수록 더 효율적인 것으로 나타났다. 따라서 본 연구에서 적용한 TS_2 방법은 지금까지 해결방법이 발표되지 않는 동일하지 않는 병렬기계 시스템에서 지연작업수를 최소화하는 하나의 해결방법으로 적합할 것이다.

참 고 문 헌

[1] 전태웅, 강맹규, "동일한 병렬기계 시스템에서 평균 지연시간의 최소화를 위한 Tabu Search 방법," 공업경영학회지, 제18권 35집, pp. 107-114, 1995.

[2] Baker, K. R., ***Introduction to Sequencing and Scheduling***, John Wiley & Sons, Inc., 1974.

[3] Barnes, J. W. and M. Laguna, "Solvng the Multipul-Machine Weighted Flow Time Problem Using Tabu Search," ***IIE Transactions***, Vol. 25, No. 2, pp. 121-127, 1993.

[4] Barnes, J. W. and J. B. Chambers, "Solving the Job Shop Scheduling Problem with Tabu Search," ***IIE Transactions***, Vol. 27, pp.257-263, 1995.

[5] Cheng, T. C. E. and C. C. S. Sin, "A State of the Art Review of Parallel Machine Scheduling Research," ***European Journal of Operatonal Research***, Vol. 47, pp.271-292, 1990

[6] Dell'Amico, M. and M. Trubian, "Applying Tabu Search to the Job-Shop Scheduling Problem," ***Annals of Operations Research***, Vol. 41, pp. 231-252, 1992.

[7] Ho, J. C. and Yih-Long Chang, "Heuristic for Minimizing Mean Tardiness for m Parallel Machines," ***Naval Research Logistics Quarterly***, Vol. 38, pp. 367-381, 1991.

[8] Ho, J. C. and Yih-Long Chang, "Minimizing the Number of Tardy Jobs for m Parallel Machines," ***European Journal of Operational Research***, Vol. 84, pp. 343-355, 1995.

[9] Laguna, M. J. Barnes, and F. Glover, "Tabu Search Methods for a Single Machine Scheduling Problem," ***Journal of Intelligent Manufacturing***, Vol. 2, No. 2, pp. 63-74, 1991.

[10] Lawler, E. L. and C. U. Martel, "Preemptive Scheduling of Two Uniform Machines to Minimize the Number of Late Jobs," ***Operations Research***, Vol. 37, No. 2, pp. 314-318, 1989.

[11] Moore, J. M., "An N-Job, One Machine Sequencing Algorithm for Minimizing the Number of Late Jobs," ***Management Science***, Vol. 15, No. 1, pp. 102-109, 1968.

[12] Taillard, E., "Some Efficient Heuristic Methods for the Flowshop Sequencing Problem," ***European Journal of Operational Research***, Vol. 47, pp. 65-74, 1990.

[13] Villarreal, J. V. and R. L. Bulfin, "Scheduling a Single Machine to Minimize the Weighted Number of Tardy Jobs," ***IIE Transactions***, Vol. 15, pp. 337-343, 1983.