

서로 다른 납기를 갖는 작업에 대한 이중 병렬기계에서의 일정계획수립

Scheduling Jobs with different Due-Date on Nonidentical Parallel Machines

강용혁* · 이홍철* · 김성식*

Yong-Hyuk Kang* · Hong-Chul Lee* · Sung-Shick Kim*

Abstract

This paper considers the nonidentical parallel machine scheduling problem in which n jobs having different due dates are to be scheduled on m nonidentical parallel machines. For the make-to-order manufacturing environment, the objective is to minimize the number of tardy jobs. A 0-1 nonlinear programming model is formulated and a heuristic algorithm that allocates and sequences jobs to machines is developed. The proposed algorithm makes use of the concept of assignment problem based on the suitability measure as the cost coefficient. Computational experiments show that the proposed algorithm is superior to the existing one in some performance measures such as number of tardy jobs. In addition, this algorithm is appropriate for solving real industrial problems efficiently.

1. 서론

주문생산(Make-to-order)환경에서는 동일한 기계들로 이루어진 작업장에서 납기가 서로 다른 여러 개의 작업에 대해 일정계획을 수립하는 문제가 많이 발생한다. 또한 각각의 기계들은 모델에 따라 또는 기계의 노후화 정도에 따라 효율성에서 서로 차이를 보인다[18]. 만일 작업장이 보유하고 있는 작업능력이 부족하여 납기지연(Tardiness)이 발생하면, 납기지연에 따른 경제적 손실과 재공재고의 증가에 따른 제조원가의 상승 및 고객만족도 저하로 인한 시장 점유율 감소를 초래한다. 따라서, 납기를 맞출 수 없는 작업은 많은 비용을 감수하면서 외주

생산을 통하여 납기를 맞추게 된다. 현실적으로 대부분의 공장에서는 일정계획을 비효율적으로 수립하기 때문에 보유하고 있는 작업능력을 가지고 충분히 납기를 맞추어 생산할 수 있음에도 불구하고 외주를 주게되는 경우가 자주 발생한다. 따라서 작업장의 능력을 충분히 활용하여 납기를 만족시키지 못하는 작업의 수를 최소화하기 위한 효율적인 일정계획 수립이 총처리시간(Makespan)이나 평균처리시간(Mean Flowtime)의 최소화보다 현장에서 더 중요하게 취급되고 있다.

병렬기계로 이루어진 작업장에서 납기에 맞추어 생산하는 것이 중요한 문제가 되는 작업장의 대표적인 형태로 플라스틱 사출공장을 들 수 있다. 플라스틱 사출공장

* 고려대학교 산업공학과

은 여러 대의 동일한 사출기에서 여러 종류의 사출품들이 만들어지고, 이 사출품들은 조립라인으로 보내지거나 바로 주문자에게 보내진다. 각각의 제품은 사출기계의 노후화 정도에 따라 생산 시간이 달라지게 된다. 조립라인에서 일정계획이 수립되면 이 계획이 차질 없이 수행되도록 사출 작업장에서 생산되는 사출품은 일정 시간까지 생산이 이루어져야한다. 또한 사출품에 대해 주문이 들어오는 경우도 역시 납기를 맞춰 생산이 이루어져야한다. 만일 사출 작업장에서 납기 내에 사출품을 생산하지 못할 경우 해당제품에 대해 많은 비용을 감수하면서 외주를 주게된다. 이와 같은 상황에서는 납기를 맞추지 못하는 작업을 최소화 시켜주도록 일정계획을 수립하는 것이 중요하다.

병렬기계에서 작업들의 납기를 고려한 일정계획에 대해서는 많은 연구가 이루어졌으나 대부분의 경우 평균 납기 지연 시간(Mean Tardiness)을 최소화시키거나 또는 납기보다 늦게 생산되는 경우(Tardiness)와 납기보다 일찍 생산 완료되는 경우(Earliness) 발생하는 비용을 최소화시키는 것을 목적으로 하고 있다.

본 논문은 작업능력에서 서로 차이가 나는 병렬 기계(nonidentical parallel machine)에 대해 납기가 서로 다른 제품들을 생산할 때 납기를 맞추지 못하는 제품의 수를 최소화 하기 위해 할당문제를 이용해 일정계획을 수립하는 알고리즘을 제시하였다.

본 논문의 구성은 다음과 같다. 제 2장에서는 관련된 기존 연구에 대해서 고찰하고, 제 3장에서는 이종의 병렬 기계에서 일정계획을 수립하는 문제를 기술하며, 제 4장에서는 문제해결을 위한 해법절차를 제시한다. 그리고 제 5장에서는 수치예제 및 실험을 통하여 제안된 해법의 성능을 기존의 해법과 비교 분석하고, 제 6장에서는 결론을 제시한다.

2. 기존연구의 고찰

지금까지 연구된 병렬기계에서 일정계획을 수립하는 문제는 작업에 대한 총처리시간(Makespan)을 최소화하는 문제, 납기를 고려한 Earliness/Tardiness 문제와 납기를 맞추지 못하는 작업의 수를 최소화하는 문제로 분류된다. Cheng과 Sin[4]은 병렬기계에서 일정계획을 수립하는 문

제를 다룬 기존의 논문들을 목적함수별로 정리해 놓았다. 기계의 작업능력이 동일하다는 가정 하에서 일정계획을 수립하는 문제에서 Karp[12]는 두 개의 동일한 기계에서 작업의 총처리시간(Makespan)을 최소화시키기 위한 일정계획 문제가 NP-hard임을 보이고 있다. Hariri와 Potts[9]는 기계의 능력이 서로 다른 병렬기계에서 총처리시간을 최소화시키는 일정계획을 세우기 위해 첫 번째 단계에서 선형계획법으로 몇몇 작업에 대해 기계에 작업을 할당하고 두 번째 단계에서 나머지 작업에 대해서 발견적 해법을 이용해 기계에 할당하는 두단계(two-phase)알고리즘에 대해서 고찰하였다. Ibarra와 Kim[11]은 작업의 총처리시간을 최소화시키는 일정계획 문제에서 기계가 두 개 존재할 때 빠른 시간 내에 좋은 해를 발견할 수 있는 알고리즘을 제시하였고, 여러 대의 기계가 존재할 때 모든 기계의 수행 능력이 동일하고 작업들간의 가공시간의 차이가 크지 않다는 조건이 주어질 때 작업의 수가 많아질수록 최적해에 가까워지는 일정계획방법에 대해서도 고찰하였다. Horowitz와 Sahni[10]는 dynamic programming 기법을 통해 최적의 총처리시간을 갖는 일정계획을 구하는 방법과 최적의 총처리시간에 근사한 일정계획을 구하는 알고리즘을 제시하고 있으나 계산량이 너무 많아 현실적이지는 못하다.

작업의 납기를 고려한 연구는 주로 작업완료 시간(Completion Time)과 납기사이의 차이를 줄이는 문제, 납기를 맞추지 못하는 작업의 수를 줄이는 문제를 다루고 있다. 또한 작업의 납기는 모든 작업에 대해서 동일한 경우와 각각의 작업의 납기가 서로 다른 경우에 대해서 연구가 이루어지고 있다. 단일기계에서 공통의 납기를 갖는 작업들에 대해서 Swzarc[17]는 Branch & Bound기법을 이용하여 작업완료 시간과 납기의 차를 줄이는 일정계획 알고리즘을 제시하였고, 김석준,이채영[1]은 유전알고리즘을 이용하여 일정계획을 수립하는 방법을 제시하였다. Wilkerson과 Irwin[19], Dagramici와 Surkis[5]는 납기가 빠른 순으로 일정계획을 세우는 규칙(EDD: Early Due Date)과 납기와 가공시간사이의 차이(Slack), 가공시간이 짧은 순서대로 일정계획을 세우는 규칙(SPT: Shortest Processing Time)을 확장하여 작업지연시간을 줄이는 방법을 제시하고 있다. Wilkerson과 Irwin[19]은 단일기계에서 납기가 서로 다른 작업에 대해서 EDD규칙을 확장

하여 평균 납기자연시간을 최소화하는 알고리즘을 제시하였고, Dagramici와 Surkis[5]는 Wilkerson과 Irwin[19]의 알고리즘을 보완하여 수행능력이 동일한 병렬기계에서 일정계획을 수립하는 방법을 제시하였다. Guinet[7]는 Dagramici와 Surkis[5]의 알고리즘을 확장하여 동일한 병렬기계로 이루어진 방직공장에서 평균 납기자연시간을 최소화시키는 일정계획을 수립하는 알고리즘을 개발하였다. Guinet[8]는 또한 Simulated Annealing 기법을 자신의 이전 알고리즘을 보완하였다. 과거에는 납기보다 늦게 생산되는 경우(Tardiness)만을 고려하였으나 최근에는 납기보다 일찍 생산되었을 경우(Earliness)에 발생하는 비용을 고려하는 문제들에 대한 연구가 활발히 이루어지고 있다. Abdul-Razaq와 Potts[2], Grey외[6], Ow와 Morton[16]은 납기보다 일찍 생산 완료했을 경우와 늦게 생산 완료했을 경우 생기는 비용의 합을 최소화시키는 발견적 알고리즘을 제시하였다.

납기를 맞추지 못하는 작업의 수를 고려한 연구는 많지 않으나, 현실적으로 납기를 맞추지 못하는 작업에 대해 대부분 외주에 의존해야만 하는 상황에서 납기를 맞추지 못하는 작업의 수를 줄이는 일정계획 방법을 제시하는 것은 중요하다. Kise[13]는 병렬기계에서 납기를 맞추지 못하는 작업의 수를 최소화하는 문제가 NP-complete임을 보이고 있다. Moore[15]는 단일기계에서 서로 다른 납기를 갖는 작업들에 대해서 납기를 맞추지 못하는 작업의 수를 최소화시키는 일정계획을 수립하는 알고리즘을 제시하였으며, Kise외[13]는 작업가용시간이 고려되었을 때 납기를 맞추지 못하는 작업의 수를 최소화시키는 알고리즘을 제시하였다. Bedworth와 Bailey[3]가 Moore[15]의 알고리즘을 변형하여 병렬기계에서 납기를 맞추지 못하는 작업의 수를 최소화시키는 일정계획 알고리즘을 제시하고 있으나 기계의 능력이 동일하다는 가정을 두고 있다. 또한 이 알고리즘은 대상작업에 대해서 납기가 빠른 순서대로 하나의 작업만을 고려하여 일정계획을 세우기 때문에 여러 작업에 대해 한꺼번에 일정계획을 수립하는 것보다 좋은 결과를 얻기 힘들다. Li[14]는 Kise외[13]의 알고리즘을 작업능력이 동일한 병렬기계에서 일정계획을 수립하는 알고리즘으로 확장하였다. Woolsey[20]는 Bedworth와 Bailey[3]가 그들의 저서에서 여러 가지 목적함수에 대해서 제시했던 병렬기계에서의 일정계획 수

립방법을 정리하였다.

본 연구에서는 작업능력이 서로 다른 병렬기계에서 납기가 서로 다른 작업에 대해서 납기를 맞추지 못하는 작업의 수를 최소화시키는 방법을 제시한다. 또한, 제안된 알고리즘의 우수성을 보여주기 위하여 작업능력이 서로 다른 병렬기계에서 일정계획을 수립할 수 있도록 Bedworth와 Bailey가 Moore의 알고리즘을 수정하여 제시한 알고리즘과 서로 비교하였다.

3. 문제정의

3.1 문제의 정의 및 가정

본 연구의 대상은 n 개의 작업에 대한 m 개의 작업능력이 서로 다른 병렬기계로 이루어진 작업장에서의 일정계획 수립이다. 각각의 작업은 서로 다른 납기를 가지고 있으며 하나의 기계에 할당되며 할당되는 기계에 따라 가공 시간이 서로 달라진다. 만일 작업 $i(i=1,2,\dots,n)$ 가 기계 $j(j=1,2,\dots,m)$ 에 할당되었을 경우 가공 시간은 p_{ij} 로 정의된다. 각 기계는 한번에 하나의 작업만을 처리할 수 있다. 각각의 기계는 기존에 할당된 작업을 가지고 있다. 즉 새로운 작업에 대해 일정계획을 수립할 때 기계 가용 시간이 현재시점이라는 가정을 하지 않는다. 또한 본 논문에서는 다음과 같은 사항을 가정하고 있다.

- ① 작업변환 시간은 작업의 가공시간에 포함되어 있는 것으로 가정한다.
- ② 각 기계는 하나의 작업에 대해서 가공이 완료될 때까지 다른 작업을 할 수 없다.
- ③ 작업사이에 선후관계가 존재하지 않는다.
- ④ 각 기계의 임시저장장소의 크기는 고려하지 않는다.

①에서 작업변환시간은 작업을 수행하는 순서에 영향을 받지 않는다. 즉, 작업변환 시간은 작업마다 일정하므로 가공시간에 포함시켜도 현실적으로 문제가 되지 않는다.

만일 작업장에서 납기를 맞추지 못하는 작업의 발생할 경우 해당 작업은 외주를 주게된다. 본 논문에서 제시한 일정계획 알고리즘은 가능한 한 많은 작업에 대해 납기를 맞춰 줄으로써 외주로 인해 발생하는 비용을 줄이기

위해 목적함수는 납기를 맞추지 못하는 작업의 수를 최소화시키는 것이다.

3.2 수리적 모델

위와 같은 문제에 대한 기호 및 수리모형은 다음과 같다.

기호 정의

- n : 작업의 수 ($i = 1, \dots, n$)
- m : 기계 수 ($j = 1, \dots, m$)
- d_i : 작업 i 의 납기
- p_{ij} : 작업 i 를 기계 j 에서 가공할 때 가공시간
- s_j : 기계 j 의 가용시간(새로운 일정계획을 수립하는 시점부터 s_j 시간 후 까지는 기존에 일정계획이 수립되어 있음을 나타낸다.)
- M : 무한히 큰 수
- y_{ij} : 기계 j 에서 작업 i 의 가공 시작 시간

$$\xi_{ik} = \begin{cases} 1 & \text{if 기계} j \text{에서 작업} i \text{가 작업} k \text{가 완료된 후 시작되면} \\ 0 & \text{if 기계} j \text{에서 작업} i \text{가 작업} k \text{가 완료된 후 시작되면} \end{cases}$$

$$x_{ij} = \begin{cases} 1 & \text{if 작업 } i \text{가 기계 } j \text{에 할당되었을 경우} \\ 0 & \text{otherwise} \end{cases}$$

$$I_i = \begin{cases} 1 & \text{if 작업 } i \text{가 납기를 맞추지 못하는 경우} \\ 0 & \text{otherwise} \end{cases}$$

수리모형

$$\text{Min } \sum_{i=1}^n I_i \tag{1}$$

subject to

$$\sum_{j=1}^m x_{ij}(y_{ij}+p_{ij})-d_i \leq M \times I_i \quad i = 1, \dots, n \tag{2}$$

$$x_{ij}x_{ik}(y_{ij}-y_{ik}-p_{ij}) \geq -(1-\xi_{ik}) \times M \quad i \neq k = 1, \dots, n \tag{3}$$

$$j=1, \dots, m$$

$$x_{ij}x_{ik}(y_{ij}-y_{ik}-p_{ij}) \geq -\xi_{ik} \times M \quad i \neq k = 1, \dots, n \tag{4}$$

$$j = 1, \dots, m$$

$$\sum_{j=1}^m x_{ij}y_{ij} \geq \sum_{j=1}^m x_{ij}s_j \quad i = 1, \dots, n \tag{5}$$

$$\sum_{j=1}^m x_{ij} = 1 \quad i = 1, \dots, n \tag{6}$$

$$I_i \in \{0, 1\} \quad i = 1, \dots, n \tag{7}$$

$$x_{ij} \in \{0, 1\} \quad i = 1, \dots, n \tag{8}$$

$$j = 1, \dots, m$$

$$\xi_{ik} \in \{0, 1\} \quad i \neq k = 1, \dots, n \tag{9}$$

$$j = 1, \dots, m$$

수리모델에서 식(1)은 납기를 맞추지 못하는 작업의 수를 최소화하는 목적함수를 나타낸다. I_i 는 작업 i 가 납기를 맞추지 못하면 1 그렇지 않으면 0을 갖는다. (2)는 작업 i 가 할당된 기계에서 작업 완료시간이 작업의 납기보다 크면 I_i 는 1이라는 값을 갖게되고 그렇지 않으면 0이라는 값을 갖도록 만들어 주므로써 목적함수(1)에서 I_i 의 값을 최소화하는 것이 납기를 맞추지 못하는 작업의 수를 최소화하도록 해준다. 식(3),(4)은 각 기계에 할당된 작업들에서 하나의 작업이 완전히 끝나야 다음 작업을 시작할 수 있다는 것을 의미한다(Nonpreemptive). ξ 가 1의 값을 갖는 경우 식(4)가 의미 없는 식이 되고, 0의 값을 갖는 경우 식(3)이 의미 없는 식이 된다. 식(5)는 각 기계에 할당된 작업들이 작업가능 시간이 현재 시점이 아닌 기존에 할당된 작업이 모두 완료되는 시점 이후라는 것을 의미한다. 식(6)은 하나의 작업은 하나의 기계에만 할당된다는 조건을 나타낸다. 식(7),(8),(9)는 0 또는 1의 값을 갖는 정수 조건이다.

4. 해법 절차

4.1 기본 개념

본 논문에서 제시하는 일정계획방법은 대상작업을 납기가 짧은 순서대로 나열한 후, 주어진 기계의 대수만큼의 작업을 선택하여 할당문제(Assignment Problem)를 이용하여 작업을 기계에 할당하며 각 기계에 할당된 작업들의 할당된 순서가 곧 그 기계에서의 작업순서가 된다. 즉, m 개의 기계와 n 개의 작업이 주어졌을 때 먼저 주어진 n 개의 작업에 대해 납기가 빠른 순서대로 나열한 후 순서대로 m 개의 작업을 선택한다. 이때 이 m 개의 작업을 m 개의 기계에 할당하는 문제로 보고 할당문제를 푼다.

할당문제에서 기계와 작업사이의 비용(c_{ij})을 나타내는 지수로는 다음과 같이 정의된 값을 이용한다.

$$c_i = p_j / (d_i - t_j) \quad (10)$$

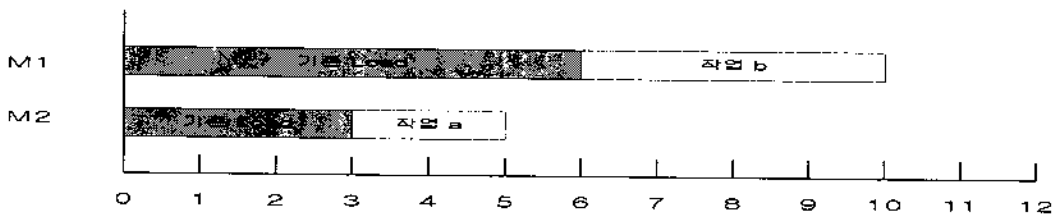
식(10)에서 t_j 는 기계 j 에서 새로운 작업에 대해 가장 빨리 가공을 시작 할 수 있는 시간을 나타내며 작업이 기계에 할당됨에 따라 변하는 값이다. 긴급율(CR: critical ratio)의 역수인 c_i 는 어떤 작업이 어느 기계에 할당하는 것이 적합한가를 나타내는 적합도(suitability index)로 쓰이며, 기계 j 에 작업 i 를 할당할 때 납기를 얼마나 잘 맞출 수 있는가를 나타낸다.

본 논문에서 제시한 알고리즘은 크게 다음의 세단계로 구성된다.

- ① 작업을 납기순으로 나열한다.
- ② 기계의 대수만큼 작업을 선택하여 작업과 기계간에 할당문제를 만든 후 할당문제를 푼다.
- ③ 할당문제를 푼 결과 납기를 만족시키지 못하는 작업이 생기면 그 작업은 할당을 제외시키고 나머지 작업들만 할당을 시킨다. 다음으로 할당에서 제외된 작업을 포함하여 아직 할당되지 않고 남아있는 작업들 중에서 기계의 대수만큼 작업을 선택후 다시 할당문제를 푼다.

Hungarian방법[21]을 이용하여 할당문제를 풀면 각 작업은 어느 기계에 할당이 되어야 하는지를 알 수 있는데, 만일 해당기계에 작업이 할당되었을 때 납기를 맞추지 못하는 작업이 있으면 그 작업은 기계 할당에서 제외시키고, 그 작업을 포함해서 아직 할당되지 않고 남아있는 작업들 중에서 m 개의 작업을 선택후 다시 할당문제를 푼다. 할당문제는 c_i 값이 작은 것을 우선으로 하나의 기계에 하나의 작업만을 할당하는데, 하나의 기계에 두 개 이상의 작업이 할당되었을 때 더 좋은 결과가 나올 수 있다. 따라서 납기를 맞추지 못하는 작업에 대해서는 할당에서 제외하고 다음번 할당문제에 포함시키므로써 다른 기계에 할당될 수 있는 기회를 주게되므로 더 좋은 일정계획을 얻을 수 있다. 예를 들어 작업a, b의 가공시간과 납기가 각각 (2, 6), (4, 9)이고 두 대의 기계(M1, M2)가 존재하고 기계의 가용시간이 각각 6, 3이라고 가정하자. 이때, 단순히 할당 문제를 풀게 되면 작업b는 기계 M1에 할당되고 완료시간은 10이 되고, 작업a는 기계 M2에 할당되고 완료시간은 5가 된다. 즉, 작업b는 납기를 맞출 수 없게 된다. 이 경우 작업b를 할당에서 제외시키고, 다시 할당문제를 만들어 풀게 되면 작업b는 M2에 할당되며 완료시간은 9가 된다. 따라서 작업a, b둘다 납기를 만족시

단순히 할당문제를 풀었을 경우 일정계획 결과



작업 b에 대해서 다시 할당문제를 풀었을 경우 일정계획 결과

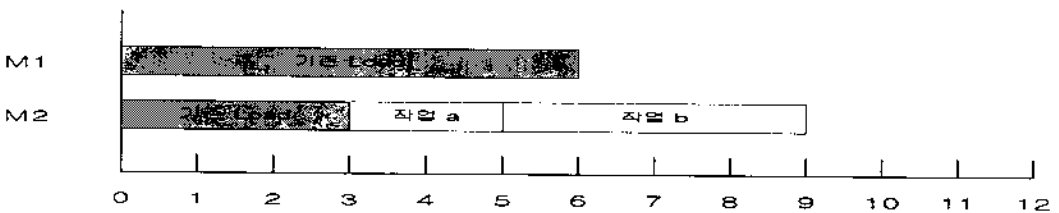


그림 1. 할당문제를 이용한 일정계획 수립 결과

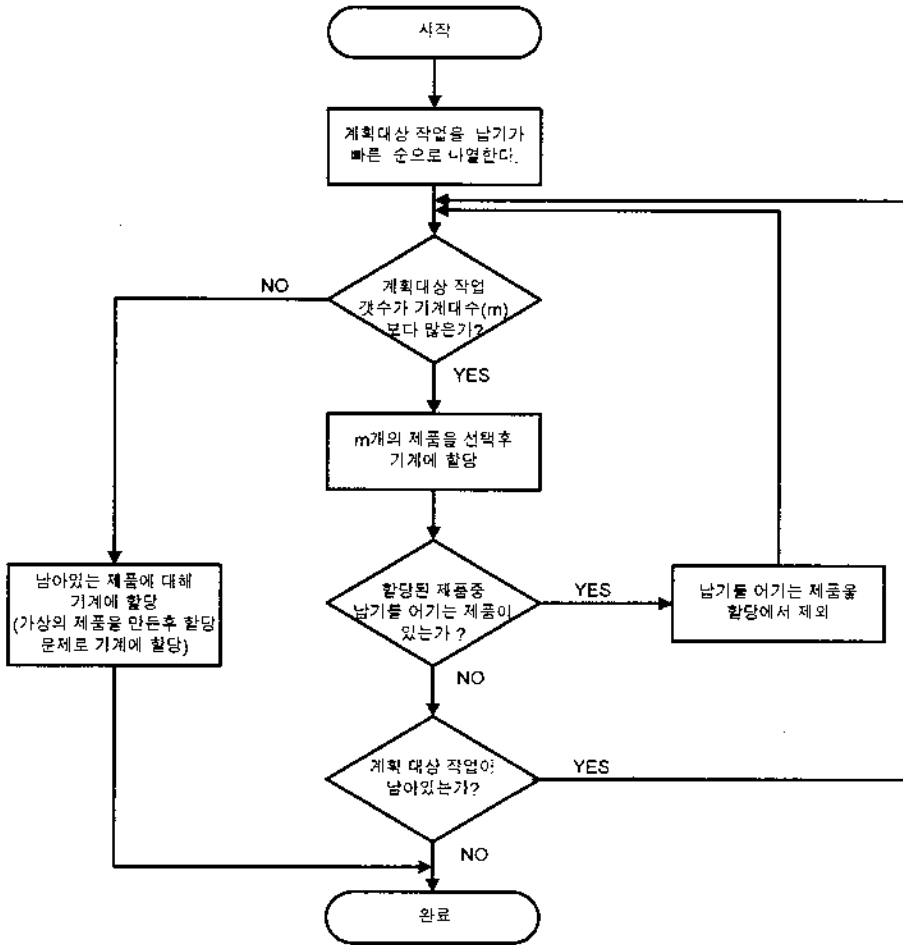


그림 2. 일정계획을 위한 순서도

키는 일정계획을 얻을 수 있다. 그림 1은 이와 같은 결과를 나타낸다. 이러한 과정이 완료되면 각 기계에 생산해야 할 작업이 할당되고, 각 작업의 작업순서가 결정된다. 그림 2는 위와 같은 절차에 대한 순서도를 보여주고 있다.

4.2 알고리즘

알고리즘 절차 및 필요한 기호는 다음과 같다.

$T = \{(p_1, \dots, p_m) \mid i = 1, \dots, n\}$: 일정계획 대상인 작업들의 전체집합

N : 남기를 맞추지 못하는 작업들의 집합

$n(T)$: 집합 T 의 원소의 개수

L_j : 기계 j 에 할당된 작업들의 집합

$L = \{L_j \mid j = 1, \dots, m\}$: 각 기계에서 일정계획의 결과

step1: L_j 와 집합 T 를 초기화한다.

1. $L_j \leftarrow \phi$ for $\forall j$

2. T 집합 내에 있는 각 작업들을 $d(i=1, \dots, n)$ 가 작은 순으로 나열한다.

step2: 알고리즘 종료 조건에 맞는지 검사한다.

If $T = \phi$ then stop

else goto step 3

step3: 할당할 작업을 선택한다.

If $n(T) \geq m$ then

집합T에서 순서대로 m개의 작업을 선택한다. goto step 4

else

goto step 5

step4: 할당문제를 이용해 작업을 기계에 할당

1. $c_j = p_j / (d - t_j)$ 를 계산한 후 할당문제를 이용해 m개의 작업을 m개의 기계에 할당한다. (단, $d_j - t_j < p_j$ 인 경우 c_j 는 M으로 정의한다.(M은 무한히 큰수))

2. $t_j \leftarrow t_j + \{j\}$ 기계에 할당된 작업의 가공시간} for $\forall j$

3. For $k \leftarrow 1$ to m step 1 do

If $t_j \leq j$ 기계에 할당된 작업의 납기 then

$T \leftarrow T - \{j\}$ 기계에 할당된 작업} and $L_j \leftarrow L_j \cup \{j\}$ 기계에 할당된 작업} for $\forall j$

else

$t_j \leftarrow t_j - \{j\}$ 기계에 할당된 작업의 가공시간} for $\forall j$

4. goto step 2

step5: 남아있는 작업이 기계대수 보다 작을 때 작업을 기계에 할당후 종료

1. $m - n(T)$ 개의 가상 작업을 만든다.

2. $c_j = p_j / (d - t_j)$ 를 계산한 후 할당문제를 이용해 m개의 작업을 m개의 기계에 할당한다.(단, $d_j - t_j < p_j$ 인 경우 c_j 는 M으로 정의하고 가상 작업의 경우 2M으로 정의한다.)

3. 가상 작업을 삭제한다.

4. $t_j \leftarrow t_j + \{j\}$ 기계에 할당된 작업의 가공시간} for $\forall j$

5. For $k \leftarrow 1$ to m step 1 do

$T \leftarrow T - \{j\}$ 기계에 할당된 작업} and $L_j \leftarrow L_j \cup \{j\}$ 기계에 할당된 작업} for $\forall j$

6. stop

Step 4에서 현재 할당이 안된 채로 남아있는 작업의 수가 기계 대수 보다 많으면 납기가 빠른 순으로 m개의 작업을 선택하여 할당문제를 풀게 되는데, 만일 남아있는 작업의 개수가 m보다 작은 경우 할당문제를 만들기 위해서는 가상의 작업을 만들어 주어야 한다(Step 5). Step

4에서는 할당문제를 이용해서 각 작업을 기계에 할당했을 경우 납기를 어기게 되는 작업에 대해서 할당에서 제외시켜 다음 기회에 다른 기계에 할당될 수 있는 기회를 주고 있다. 만일 남아있는 작업이 m보다 큰 경우라 할지라도 할당문제를 풀어본 결과 모든 작업이 납기를 못 맞춰 할당에서 제외되는 작업들이라면 이 작업들에 대해서 위의 알고리즘을 계속 적용할 경우 Step 3과 Step 4를 무한히 반복되게 된다. 이를 방지하기 위해서는 다음과 같은 조치를 취해준다.

먼저, 납기가 가장 긴 작업이 선택될 때까지 위의 알고리즘을 수행하는 과정에서 할당문제를 풀어본 결과 선택된 m개의 작업이 모두 납기를 맞추지 못하는 경우 그 작업들을 집합 T에서 제외시키고 집합 N에 포함시킨다. 집합 T에 남아있는 작업이 하나도 없을 때까지 위의 알고리즘을 수행하고, 집합 N에 작업이 존재하면 납기 순서대로 m개의 작업을 선택하여 할당문제를 풀고 납기에 상관없이 할당문제의 결과대로 작업을 기계에 할당한다.

5. 수행도 평가

5.1 수치예제

본 논문에서 제시한 해법의 절차를 자세히 설명하기 위해 Bedworth와 Bailey[3]가 제시한 기준문제를 변형하여 본 해법을 적용시켜 해를 도출하였다. 주어진 기본 문제에서는 10개의 작업에 대해 3개의 동일한 병렬기계에서 작업을 할당하고 작업순서를 결정하게 된다. 본 논문에서 제시한 이기종 병렬기계에 대한 조건을 만족시키기 위해 각 작업에 대한 가공시간을 기계에 따라 다르게 설정하였다. 기준문제를 위한 데이터는 표 1과 같다.

먼저 Step 1에 따라 대상작업을 납기가 빠른 순으로 나열하면 $T = \{ 6, 10, 1, 7, 2, 8, 5, 4, 3, 9 \}$ 가 된다. 기계의 수가 3이므로 순서대로 세 개의 작업을 선택한 후 (6, 10, 1) step4에 따라 $c_{61} = 0.25, c_{62} = 0.4, c_{63} = 0.6, c_{101} = 0.143, c_{102} = 0.286, c_{103} = 0.5, c_{11} = 0.5, c_{12} = 0.625, c_{13} = 0.75$ 가 된다. 기계 1, 2, 3과 작업 6, 10, 1에 대해 적합도를 최소로 만드는 할당문제를 풀면 작업 6은 기계 1에, 작업 10은 기계 2에, 작업 1은 기계 3에 할당되며, 각 기계에서 기계가용시간은 $t_1 = 2, t_2 = 2, t_3 = 8$ 이 된다. 그리고 각 작업은 납기 내에 생산 가능

표 1. 기준문제를 위한 기초 데이터

Job i	Machine1	Machine2	Machine3	Due Date d
	가공시간 p_{i1}	가공시간 p_{i2}	가공시간 p_{i3}	
1	4	5	6	8
2	5	6	7	9
3	2	3	4	14
4	7	8	9	12
5	6	7	8	11
6	1	2	3	5
7	2	3	4	8
8	4	5	6	10
9	3	4	5	15
10	1	2	3	7
기계가용시간(s_i)	1	0	0	

※ 기계가용시간 s_i 는 기계에서 새로운 작업에 대해 가공을 시작할 수 있는 시간을 나타낸다. 즉, 새로운 일정계획을 수립하는 시점부터 s_i 시간 후까지는 기존에 일정계획이 수립되어 있음을 나타낸다.

하므로 작업 6, 10, 1에 대한 일정계획이 확정된다. 즉, 작업 6은 기계 1에서 가공시작시간 = 1, 가공완료시간 = 2. 작업 10은 기계 2에서 가공시작시간 = 0, 가공완료시간 = 2, 작업 1은 기계 3에서 가공시작시간 = 0, 가공완료시간 = 6이된다. 따라서 $L_1 = \{ 6 \}$, $L_2 = \{ 10 \}$, $L_3 = \{ 1 \}$, $T = \{ 7, 2, 8, 5, 4, 3, 9 \}$ 로 갱신된다. 다음

은 작업 7, 2, 8을 선택하여 위의 과정을 반복한다. 이와 같은 절차를 거치면 그림 3과 같은 일정계획이 수립된다. $L_1 = \{ 6, 7, 8, 4 \}$, $L_2 = \{ 10, 2, 3, 9 \}$, $L_3 = \{ 1, 5 \}$ 가 된다. 작업 4, 5는 납기를 맞추지 못하고 평균 납기 지연 시간은 2.33, 최대 납기 지연 시간은 4가 된다.

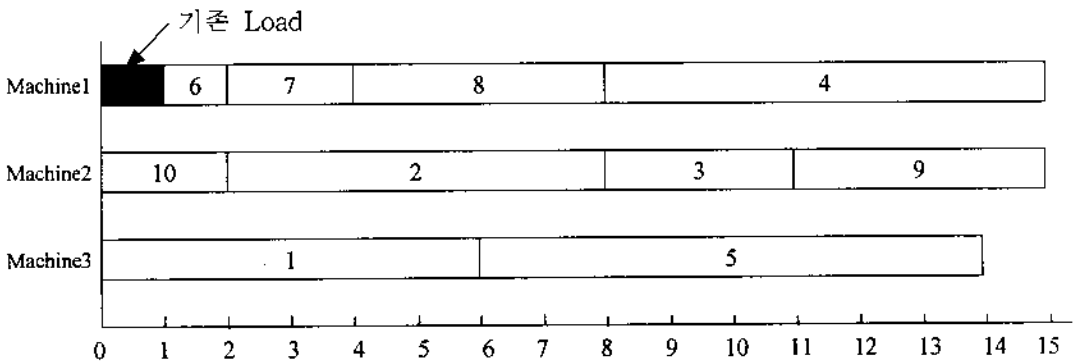


그림 3. 기준문제에 대한 일정계획 결과

5.2 실험설계

본 논문에서 제안된 알고리즘의 수행도를 평가하기 위해, 기계가 10대이고 작업의 개수가 50개인 임의의 문제를 만들고 알고리즘을 적용해 계산하였다. 각각의 기계는 기계의 노후화 정도에 따라 가공능력을 세 가지로 구분했으며, 각각 단위 가공 시간이 1, 2, 3으로 정해져있다. 또한 각각의 작업들은 배치의 크기가 서로 다르며 납기가 서로 다르다. 즉, 각 작업의 가공시간 p_{ij} 는 (i 작업의 배치의 크기 $\times j$ 기계의 단위 가공시간) 으로 정해진다. 기존에 제시된 방법중 이와 같은 문제를 해결하기 위한 가장 적절한 방법으로는 Bedworth와 Bailey[3]가 Moore[15]의 알고리즘을 확장하여 제시한 알고리즘이 있다. 이 알고리즘은 기계의 가공 능력이 일정하다(Identical machines)는 가정을 두고 있기 때문에 본 논문에서는 이 알고리즘을 본 논문에서 제시된 문제에 맞게 변형하여(non-identical machines) 본 논문의 알고리즘과 비교하였다. 변형된 Moore의 알고리즘은 다음과 같다.

Modified Moore's Algorithm

Step1. 작업을 기계에 할당 및 가공 순서 결정

납기가 빠른 순으로 작업을 선택하여 각 기계에서의 가공시간을 고려하여 가장 빨리 작업을 마칠 수 있는 기계에 할당하고 할당되는 순서가 가공순서가 된다.

Step2. 가공시간을 고려하여 가공순서 변경

1. 각기계에 대해서 처음으로 납기를 맞추지 못하는 작업을 찾아내어 그 이전의 작업중 가공시간이 가장 긴 작업을 그 기계의 작업순서 마지막으로 놓는다.
2. 더 이상 좋은 해가 안 나올 때까지 1을 반복한다.

본 연구는 각각의 작업에 대해서 납기를 맞추는 것을 주목적으로 하므로 비교를 위한 성능지표를 다음의 3가지로 정하였다.

- 1) 납기지연 작업의 수(Number of Tardy Jobs): NumT
- 2) 평균 납기 지연 시간(Mean Tardiness): MeanT
- 3) 납기 지연 시간의 최대값(Maximum Tardiness): MaxT

또한 짧은 계산 시간에 비교적 좋은 해를 구할 수 있어 작업현장에서 많이 쓰이고 있는 Dagranicici와 Surkis [5]가 제시한 평균 납기 지연 시간을 줄이는 알고리즘(mean tardiness), Bedworth와 Bailey[3]가 제시한 EDD규칙을 이용한 납기 지연시간의 최대값을 최소화시키는 알고리즘(max tardiness)을 이용하여 본 논문에서 제시된 알고리즘의 성능지표와 비교하였다.

50개의 작업으로 이루어진 작업군을 형성할 때 납기에 여유가 많은 작업군(A Group), 조금 여유가 있는 작업군(B Group), 납기를 맞추기 힘든 작업군(C Group) 세 가지 형태가 나오도록 작업 데이터를 형성하였고 각각의 경우에 대해서 비교 대상 알고리즘과 본 논문에서 제시한 알고리즘은 어떠한 결과를 보이는지를 비교하였다. 데이터 생성과 알고리즘 계산을 위하여 C++언어를 이용하여 프로그래밍 하였다. 필요한 데이터를 얻기 위해 작업의 생산량과 기계의 가공시간은 난수를 발생시켜 각각 (50, 150), (0, 300)범위를 갖는 균일 분포(Uniform Distribution)를 이루도록 생성하였고, 작업의 납기는 표 2에서와 같이 A, B, C 작업군에 따라 각각 (950, 1150), (750, 950), (650, 750)범위를 갖는 균일 분포를 이루도록 작업군을 생성하였다. 기계의 단위 가공 시간은 기계의 노후화 정도에 따라 1, 2, 3중 하나의 값을 갖도록 설정하였다. 또한 기계 대수의 증가에 따른 납기 지연작업의 수의 변화를 분석하기 위해 B작업군에 해당하는 작업에 대해서 기계대수와 작업수가 각각 (10, 50), (20, 100), (30, 150)개인 데이터를 생성한 후 결과를 분석하였다.

5.3 결과분석

각 작업군에 대해서 각각 50개의 작업과 10대의 기계로 이루어진 20개의 문제(총 60문제)를 만들어서 계산한 결과 납기 지연 작업의 수, 평균 납기 지연 시간, 납기 지연 시간의 최대값에 대한 평균은 표 3과 같다. 그리고 표 2의 예에서와 같이 납기에 여유가 많은 작업군(A), 조금 여유가 있는 작업군(B), 납기를 맞추기 힘든 작업군(C) 세 가지 형태의 문제에 대해서 각 4개씩 12개의 문제를 만들어서 일정계획을 수립하였을 때 작업군과 알고리즘에 따른 NumT, MeanT, MaxT 계산 결과는 그림 4, 그림 5, 그림 6과 같다.

표 3에 나타난 결과는 본 논문에서 제시한 알고리즘이

표 2. 작업군별 작업수량과 납기 예

작업번호	분류	A작업군		B작업군		C작업군	
		수 량	납 기	수 량	납 기	수 량	납 기
1		122	1139	85	795	82	656
2		105	1040	95	833	132	724
3		139	955	145	911	137	658
4		61	1009	106	905	107	680
5		142	1012	132	921	147	711
6		86	1025	55	776	85	729
7		100	1063	120	941	147	684
8		127	964	105	850	97	668
9		139	1123	145	762	104	667
10		106	1111	63	868	138	700
11		116	1012	62	914	147	683
12		129	989	64	872	81	682
13		63	1087	126	907	123	655
14		131	1016	143	776	72	744
15		123	1095	114	852	51	706
16		98	1126	145	860	137	683
17		137	1054	147	798	54	669
18		128	1113	130	764	72	674
19		108	1093	83	852	62	736
20		61	971	59	759	122	654
21		117	1097	76	853	57	659
22		121	1080	140	916	127	726
23		146	1044	97	819	103	727
24		92	1068	64	938	123	665
25		81	959	116	895	134	700
26		127	988	60	867	54	679
27		75	1040	70	931	68	679
28		120	1012	89	753	138	659
29		148	1092	144	769	122	678
30		123	1144	54	784	107	698
31		140	1073	54	811	67	732
32		117	1138	108	902	88	689
33		56	962	82	848	149	733
34		73	1125	52	807	81	708
35		110	1053	133	895	60	685
36		140	1149	135	853	72	733
37		122	987	63	861	137	729
38		119	1120	127	899	83	709
39		89	970	137	830	64	702
40		141	1013	108	947	56	737
41		105	1084	94	946	82	656
42		55	1100	104	914	86	720
43		127	950	89	949	81	684
44		54	1031	74	828	114	720
45		64	973	50	884	65	655
46		102	1072	140	865	140	748
47		56	1042	68	820	119	742
48		56	952	118	909	67	656
49		125	1128	103	889	97	653
50		122	1059	51	824	75	671

표 3. 알고리즘 적용 결과 테이블

상능지표 \ 알고리즘	Modified Moore's Algorithm	Dagramici & Surkis's Algorithm	Bedworth & Bailey's Algorithm	Proposed Algorithm
NumT(개)	12.10	16.10	16.10	10.20
MeanT(시간)	69.07	60.56	64.12	67.33
MaxT(시간)	489.20	374.60	366.6	529.50

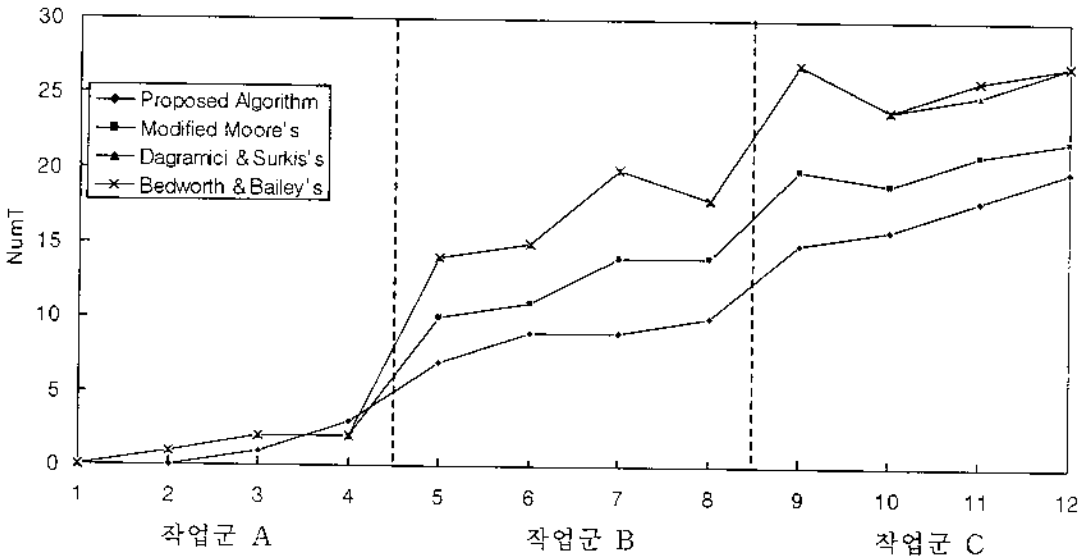


그림 4. 작업군별 납기 지연 작업의 수

납기 지연 작업의 수에서 비교대상 알고리즘들보다 좋은 결과를 얻고 있다는 것을 보여준다. 또한 같은 목적함수를 갖는 변형된 Moore의 알고리즘에 비해 납기 지연 작업의 수뿐만 아니라 평균 납기 지연 시간도 좋은 결과를 얻고 있다. 각각의 알고리즘은 본래 그 알고리즘에서 목적함수로 사용되는 값에서 우수함을 보이고 있다. 납기 지연 작업의 수를 최소화시키는 것을 목적함수로 하고 있는 변형된 Moore의 알고리즘과 본 논문에서 제시된 알고리즘은 납기 지연 작업의 수에서는 비교대상이 되는 다른 알고리즘에 비해 월등히 우수한 반면, 납기 지연 시간의 최대값은 다른 알고리즘에 비해 아주 좋지 못한 결과를 얻었다. 이러한 결과는 한정된 작업능력을 갖고 있는 작업장에서 납기가 지연되는 작업의 수를 줄이기 위해 보다 효율적으로 작업을 기계에 할당할 결과 몇몇 작업에

대해 납기 지연 시간이 커지기 때문에 발생하는 결과라고 볼 수 있다.

납기의 여유 정도에 따른 일정계획 결과를 보면, 그림 4에서 납기에 여유가 없을수록 본 논문에서 제시한 알고리즘이 비교대상 알고리즘들보다 더 나은 결과를 보인다. C작업군에 대해서 변형된 Moore의 알고리즘은 평균 20.5개의 납기 지연 작업이 생기는 반면 본 논문에서 제시된 알고리즘에서는 17.25개에 불과하다. 이는 본 논문의 알고리즘이 여러 개의 작업(기계대수 만큼의 작업)에 대해 한꺼번에 작업할당을 고려하는 반면, 변형된 Moore의 알고리즘에서는 한 번에 하나의 작업만을 할당하고 있기 때문에 뒤따르는 작업이 그만큼 납기를 맞추기 힘들어지는 상황이 발생하므로 생기는 결과라고 볼 수 있다. 실제로 기계의 대수를 늘려서 계산했을 때 납기 지연 작업의

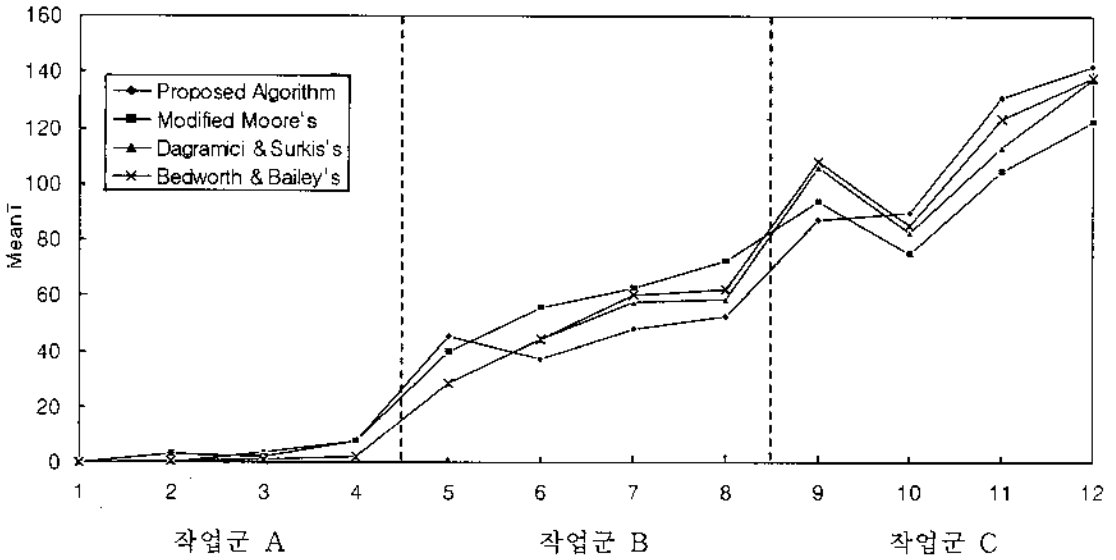


그림 5. 작업군별 평균 납기 지연 시간

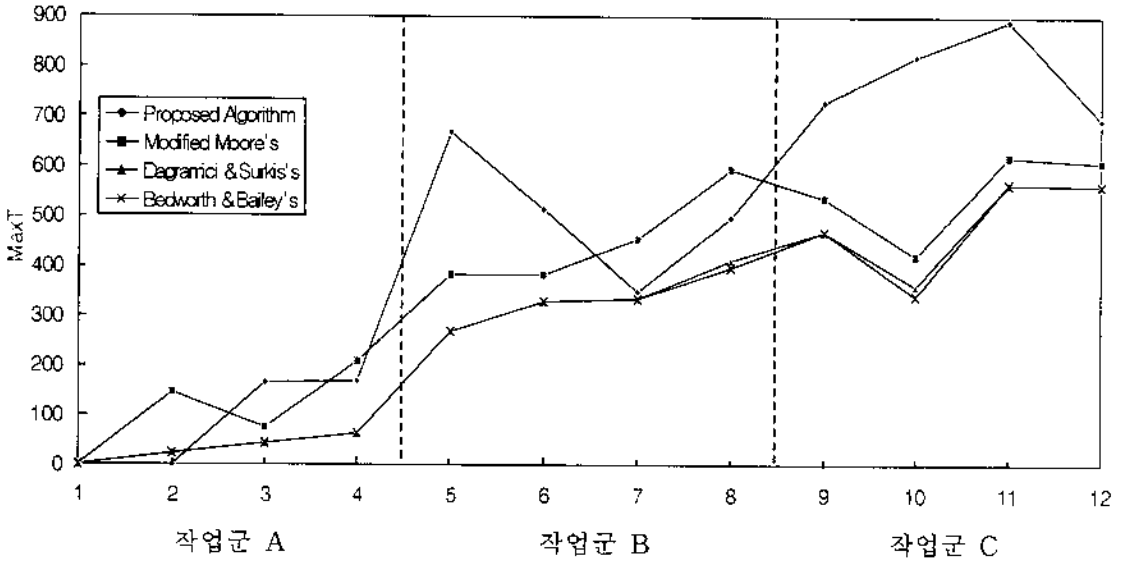


그림 6. 작업군별 최대 납기 지연 시간

수는 더욱더 차이가 난다. 표 4에서 기계의 대수가 증가함에 따라 더 좋은 결과를 얻고 있음을 알 수 있다. 그림 5에서 평균 납기 지연 시간도 납기의 여유가 적어짐에 따라 본 알고리즘이 변형된 Moore의 알고리즘보다 좋은 결과를 나타내고 있고, B작업군에 대해서는 평균납기·지연

시간을 최소화시키는 알고리즘보다 더 좋은 결과를 얻고 있다. 그림 6에서 최대 납기 지연 시간은 납기의 여유 정도에 상관없이 변형된 Moore의 알고리즘이 좋게 나오고 있다.

표 4. 기계 대수의 변화에 따른 성능지표 비교 테이블

구분	알고리즘	Modified Moore's Algorithm			Proposed Algorithm		
		NumT	MaxT	MeanT	NumT	MaxT	MeanT
기계10, 작업50		11	179	39.4	8	239	39.58
기계20, 작업100		21	442	41.36	14	857	36.74
기계30, 작업150		31	381	40.56	20	902	47.68

6. 결론

본 논문에서는 수행능력이 서로 다른 병렬기계에서 납기보다 늦게 생산되는 작업의 수를 최소화시키기 위하여 할당문제를 이용하여 일정계획수립을 수립하는 방법에 대한 알고리즘을 제시하였다. 본 논문에서 제시한 알고리즘에서는 주어진 문제를 할당문제로 변형하기 위해 기계의 가용시간, 작업의 가공 시간, 납기를 이용해 c_i 값을 얻은 후 이 할당문제를 해결하므로써 여러 개의 작업에 대해서 동시에 일정계획을 세울 수 있다. 이와 같이 여러 작업에 대해 동시에 일정계획을 수립하므로써 기존에 제시된 알고리즘에 비해 납기 지연 작업의 수를 상당히 감소시킬 수 있다. 특히 수치예제에서 보여지듯이 주어진 작업에 비해 작업능력이 모자라는 경우, 기계의 대수가 많은 경우에 납기 지연 작업의 수가 더욱 감소한다는 사실을 알 수 있다. 본 논문에서 제시한 알고리즘을 수행한 후 얻어진 각각의 기계에 대한 일정계획을 Moore의 알고리즘 Step 2(작업의 가공시간을 고려하여 재일정계획)를 적용할 경우 조금더 나은 결과를 얻을 수 있다.

본 논문에서 제시한 알고리즘을 플라스틱 사출공장과 같이 작업장이 능력이 서로 다른 병렬기계로 이루어져 있고, 작업에 대한 납기를 맞추는 것이 중요한 제조업체에서 실제 현장에 적용하여본 결과 기존의 일정계획 수립 방법에 비해 훨씬 많은 작업에 대해 납기를 맞출 수 있었다.

참고문헌

- [1] 김석준, 이채영, "작업 일정계획문제 해결을 위한 유전알고리즘의 응용", 한국경영과학회지, 제17권, 제3호, pp.1-12, 1992.
- [2] Abdul-Razaq, T. S. and Potts, C. N. "Dynamic Programming State-Space Relaxation for Single-Machine Scheduling", *J. Opnl. Res. Soc.* Vol. 39, No. 2, pp.141-152, 1988.
- [3] Bedworth, D. D. and Bailey, J. E. "Integrated Production Control Systems", John Wiley & Sons, Inc. New York, 1987
- [4] Cheng, T. E. C and Sin, C.C.S. "A State-of-the-Art Review of parallel-Machine Scheduling Research", *European J. of O.R.*, Vol. 47, No. 3, pp.271-292, 1990.
- [5] Dagramici, A. and Surkis, J. "Scheduling Independent Jobs on Parallel Identical Processor", *Management Science*, Vol. 25, No. 12, 1979.
- [6] Grey, M., Tarjan, R. and Wilfong, G. "One Processor Scheduling With Symmetric Earliness and Tardiness Penalties", *Math. Opns. Res.*, Vol. 13, pp.330-348, 1988.
- [7] Guinet, A., "Textile Production Systems: A Succession of Non-identical Parallel Processor Systems", *Journal of the Operation Research Society*, Vol. 42, No. 8, pp.655-671, 1991.
- [8] Guinet, A., "Scheduling Independent Jobs on Uniform Parallel Machines to Minimize Tardiness Criteria", *Journal of Intelligent Manufacturing*, Vol. 6, No. 2, pp.95-103, 1995.
- [9] Hariri, A. M. A. and Potts, C. N. "Heuristics for Scheduling Unrelated Parallel Machines", *Computers Ops. Res.*, Vol. 18, No. 3, pp.323-331, 1991.

- [10] Horowitz, E. and Sahni, S. "Exact and Approximate Algorithms for Scheduling Nonidentical Processors", *J. ACM*, Vol. 23, No. 2, pp.317-327, 1976.
- [11] Ibarra, O. H. and Kim, C. E. "Heuristic Algorithms for Scheduling Independent Tasks on Nonidentical Processors", *J. ACM*, Vol. 24, No. 2, pp.280-289, 1977.
- [12] Karp, R. M. "Reducibility among Combinatorial Problems", *Complexity of Computer Computations* (Edited by R.E. Miller and J.W. Thatcher), pp. 85-103. Plenum Press, New York, 1972.
- [13] Kise, H. , Ibaraki, T. and Mine, H., "A Solvable Case of the One-Machine Scheduling Problem with Ready and Due Times", *Operations Research*, Vol. 26, No. 1, pp.121-126, 1978.
- [14] Li, C. L., "A Heuristic for Parallel Machine Scheduling with Agreeable Due Dates to Minimize the Number of Late Jobs", *Computers & OR*, Vol. 22, No. 3, pp.277-283, 1995.
- [15] Moore, J. M. "Sequencing n Jobs on One Machine to Minimize the Number of Tardy Jobs", *Management Science*, Vol. 17, No. 1, 1968.
- [16] Ow, P. S. and Morton, T. E. "The Single Machine Early-Tardy Problem", *Management Science* Vol. 35, pp.177-191, 1989.
- [17] Szwarc, W. "Single-Machine Scheduling to Minimize Absolute Deviation of Completion Times from a Common Due Date", *Naval Research Logistics*, Vol. 36, No. 5, pp.663-673, 1989.
- [18] Watson, E. F. and Egbelu, P. J. "Scheduling and Machining of Jobs Through Parallel Nonidentical Machine Cells", *Journal of Manufacturing Systems*, Vol. 8, No. 1, pp.59-68, 1989.
- [19] Wilkerson, J. L. and Irwin, J. D. "An Improved Method for Scheduling Independent Tasks", *AIE Transactions*, Vol. 3, No. 3, 1971.
- [20] Woolsey, R. E. D., "Production Scheduling Quick and Dirty Methods for Parallel Machines", *Production & Inventory Management Journal*, Vol. 31, No. 3, pp.84-87, 1990.
- [21] Wu. N and Coppins. R. "*Linear Programming and Extensions*", McGraw-Hill, Inc. New York, 1981.