

패밀리 셋업이 존재하는 병렬기계 일정계획 수립

권익현* · 신현준** · 엄동환*** · 김성식*

Scheduling for Parallel Machines with Family Setup Times

Ick-Hyun Kwon* · Hyun-Joon Shin** · Dong-Hwan Eom*** · Sung-Shick Kim*

Abstract

This paper considers identical parallel machine scheduling problem. Each job has a processing time, due date, weight and family type. If a different type of job is followed by prior job, a family setup is incurred. A two phased heuristic is presented for minimizing the sum of weighted tardiness. In the first phase, using rolling horizon technique, group each job into same family and schedule each family. In the second phase, assign each job to machines using schedule obtained in the first phase. Extensive computational experiments and comparisons among other algorithms are carried out to show the efficiency of the proposed algorithm.

Keyword : Family Setup, Parallel Machines, ATCS(Apparent Tardiness Cost with Setup),
Tabu Search, Total Weighted Tardiness

1. 서론

오늘날 고객의 요구는 날이 갈수록 다양해지고 있으며 이로 인하여 기업들은 과거 대량 생산체제와는 다른 새로운 환경에 직면하고 있다. 또한 제품

의 품질과 가격적인 측면이 아닌 고객의 다양한 주문에 대한 납기 준수와 공급기간 단축이 제조업체에게 보다 중요한 요인으로 부각되고 있다. 특히 납기지연에 대한 손해비용과 고객 만족도 저하는 기업의 경쟁력 약화로 이어지기 때문에 이러한 고객

논문접수일 : 2004년 3월 5일 논문게재확정일 : 2004년 11월 29일

* 고려대학교 산업시스템정보공학과

** 삼성전자

*** 육군본부

의 요구를 적시에 만족시키는 것이 기업의 성패를 좌우한다고 할 수 있다[23]. 그러나 현실적으로 제조현장의 생산능력은 한정되어 있고 자재 부족이나 생산 설비의 고장 등의 예상치 못한 요인들로 인하여 모든 작업의 납기를 만족시키는 것은 현실적으로 매우 힘들다. 뿐만 아니라 비효율적인 일정계획 수립으로 인해 주어진 생산능력조차도 최대한 이용하지 못하고 있는 경우가 대부분이다. 따라서 제조공정의 특성과 제약조건 및 작업의 우선순위 등을 잘 반영하여 공정의 생산능력을 최대한 발휘할 수 있는 일정계획을 수립하는 것이 필수적이라 하겠다.

본 논문은 패밀리 셋업(family setup)이 존재하는 동일한 병렬기계들로 구성된 작업환경을 연구 대상으로 한다. 여기서 패밀리 셋업이란 순서 의존적인 작업준비 시간(sequence-dependent setup times)의 특수한 형태로서 제품의 특성이나 작업의 종류가 상이할 경우 발생하는 작업준비 시간을 의미한다. 즉, 동일한 형태의 작업준비가 필요한 작업군(이하 패밀리) 내의 작업들 간에는 발생하지 않고 서로 다른 패밀리에 속하는 작업 간에 발생하는 가공준비 시간을 의미한다. 이러한 패밀리 셋업 시간은 일반적으로 제품의 가공 시간에 비해 매우 크다.

패밀리 셋업 시간(family setup times)이 존재하는 대표적인 예로 TFT-LCD(Thin Film Transistor Liquid Crystal Display) 제조 공정을 들 수 있다. TFT-LCD 제조 공정은 크게 Array(FAB) 공정, 셀(cell) 공정, 모듈(module) 공정으로 구성되는데, 이 가운데 모듈 공정에서 이러한 패밀리 셋업이 존재한다. 모듈 공정에서는 13", 14", 15", 17" 등의 크기를 갖는 LCD 기판이 OLB와 Tab Solder 그리고 Silicon Dispense 공정 등을 순차적으로 수행하는 여러 대의 동일한 병렬기계에서 가공되며, 가공되는 LCD 기판의 크기가 변경될 때마다 가공 시간에 비해 아주 큰 가공 준비시간이 발생한다. 이러한 복잡한 작업절차와 작업준비 시간이 요구되는 모듈 공정은 기존의 직관적이고 단순한 방법으로 효과적인 일정계획을 수립하기 어려운 실정이다.

이와 같은 패밀리 셋업이 존재하는 공정의 경우 각 병렬 라인의 생산율(throughput)을 높이기 위해서 패밀리 셋업의 횟수를 최소화하는 것이 요구된다. 그러나 앞서 언급했던 TFT-LCD 제조 공정과 같이 대부분의 주문생산(make-to-order) 환경에서는 각각의 작업들에 대한 납기가 존재하고 작업을 주문한 주문처에 따라 서로 다른 우선순위를 갖고 있기 때문에 이들 또한 반영하는 것이 필요하다. 따라서 작업들의 납기와 우선순위, 패밀리 셋업을 고려하여 전체 계획 대상 작업에 대한 납기 지연 가중치의 합(total weighted tardiness)을 최소로 하는 일정계획을 수립하는 것이 현실적이라 할 수 있겠다. 본 논문은 패밀리 셋업이 존재하는 동일 병렬기계로 이루어진 작업환경에서 납기 지연 가중치의 합을 최소화하는 일정계획을 수립하는 알고리즘을 제시하였다.

2. 기존 연구의 고찰

지금까지 연구된 병렬기계에서 일정계획을 수립하는 문제는 작업에 대한 총 처리시간(makespan)을 최소화하는 문제, 납기를 맞추지 못하는 작업의 수를 최소화하는 문제, 셋업 횟수를 최소화하는 문제, 그리고 납기를 고려한 earliness/tardiness 문제 등으로 분류된다[14].

Cheng & Sin[4]은 병렬기계에서 일정계획을 수립하는 문제를 다른 기존의 논문들을 목적함수별로 정리해 놓았다. 이러한 대부분의 병렬기계에 관한 일정계획 문제는 NP-hard로 알려져 있는데, 한 예로 Karp[12]는 기계의 작업능력이 동일하다는 가정 하에서 일정계획을 수립하는 문제에서 두 개의 동일한 기계에서 작업의 총 처리시간을 최소화시키기 위한 일정계획 문제가 NP-hard임을 보이고 있다. Hariri & Potts[10]은 기계의 능력이 서로 다른 병렬기계에서 총 처리시간을 최소화시키는 일정계획을 세우기 위해 첫 번째 단계에서 선행 계획으로 몇몇 작업에 대해 기계에 할당하고, 두 번째 단계에서 나머지 작업에 대해서 발견적 기법(heuristic)을

이용해 기계에 할당하는 알고리즘에 대해서 고찰하였다. Horowitz & Sahni[11]는 동적 계획(dynamic programming)을 통해 최적의 총 처리시간을 구하는 방법과 최적의 총 처리시간에 근사한 일정계획을 구하는 알고리즘을 제시하고 있으나 계산량이 너무 많아 현실적이지는 못하다.

작업준비 시간이 존재하지 않는 동일 병렬기계의 납기지연 가중치의 합을 최소화하는 문제를 다룬 경우로는 Elmagraby & Park[6], Barnes & Brennan[2] 그리고 Vasilescu & Amar[22] 등이 있다. 이들은 분지 한계법(branch and bound)을 이용하여 최적해를 구하는 방법을 제안하였으며 Dogramaci[5]는 동적 계획을 이용한 최적해를 구하는 방법을 제안하였다.

패밀리 셋업이 존재하는 동일 병렬기계 문제를 다룬 경우로는 Monma & Potts[16], Tang[21], Bitran & Gilbert[3] 그리고 Azizoglu & Webster[1] 등이 있다. Monma & Potts는 preemption이 허용되는 상황에서 작업완료 시간의 최대값(C_{max})을 최소화하는 발견적 기법을 제안하였으며 Tang은 동일한 패밀리 사이에서의 파트 셋업(part setup)까지 고려한 경우의 작업완료 시간의 최대값을 최소화하여 근사 최적해를 구하는 발견적 기법과 최적해에 대한 두 개의 하한(lower bound)을 제안하였다. Bitran & Gilbert는 분지 한계법을 이용하여 패밀리 셋업의 횟수를 최소화하는 방법을 제안하였다.

패밀리 셋업과 유사한 경우인 순서 의존적(sequence-dependent)인 작업준비 시간이 존재하는 동일한 병렬기계 문제를 다룬 경우로는 Ovacik & Uzsoy[17], Schutten & Leussink[20] 그리고 Lee & Pinedo[15] 등이 있다. Ovacik & Uzsoy는 병렬기계 문제를 일련의 단일기계 문제로 분할(decomposition)하고 각각의 단일기계 문제에 대해 분지 한계법을 이용하여 납기지연 시간의 최대값(L_{max})을 최소화하는 방법을 제안하였고, Schutten & Leussink는 분지 한계법을 이용하여 납기지연 시간의 최소값을 구하는 방법을 제안하였으며 Lee & Pinedo는 각각의 작업에 대한 모수(parameter)를 계산하고

이러한 모수값을 바탕으로 작업을 기계에 할당한 뒤, 시뮬레이티드 어닐링(simulated annealing) 방법을 사용하여 보다 개선된 해를 구하는 과정을 통하여 납기지연 가중치의 합을 최소화하는 방법을 제안하였다.

이상과 같이 기존 연구들을 고찰해 본 결과, 현재 까지 발표된 논문들 가운데 본 연구와 동일한 일정 계획 문제를 고려한 경우는 존재하지 않았다. 즉, 병렬기계 문제를 다룬 기존 연구가운데에서 패밀리 셋업을 가정한 논문은 많지 않으며 이러한 논문이 고려하고 있는 목적함수는 작업완료 시간의 최대값(C_{max})이나 패밀리 셋업의 횟수 등을 최소화하는 경우가 대부분이다. 그러나 보다 현실적이고 실제 생산 환경을 잘 반영하기 위해서는 납기관련 목적함수를 고려하는 것이 더 바람직할 것이고 이에 본 연구에서처럼 납기지연 가중치의 합을 최소화하는 문제에 보다 큰 의미를 부여할 수 있을 것이다.

3. 알고리즘

3.1 문제의 정의 및 가정

본 논문에서는 가공 시간(b_j), 가중치(w_j), 납기(d_j), 패밀리 셋업(family setup) 시간(S_j)을 갖는 n 개의 작업과 m 개의 동일한 병렬기계에 대한 일정계획 수립 문제를 다룬다. 모든 작업은 계획시작 시점에서 가공 가능한 것으로 가정하여 공정 가능 시간(release time)은 존재하지 않는다. 또한 본 논문에서는 다음과 같은 사항을 추가적으로 가정하고 있다.

- 하나의 작업을 복수개의 작업으로 분할(job split)하여 할당하는 것은 불가능하다.
- 각 기계는 하나의 작업에 대해서 가공이 완료될 때까지는 다른 작업을 할 수 없다(nonpreemptive).
- 각 기계의 입시 저장장소의 크기는 고려하지 않는다.

본 논문에서는 이러한 주어진 상황과 가정 하에서 납기 지연 가중치의 합을 최소화하는 일정 계획을 수립하는 알고리즘을 제시한다.

3.2 접근 방법

앞서 기존 연구를 살펴보면 병렬기계 문제를 푸는 해법은 크게 두 가지 방법으로 요약되는데, 첫 번째 방법은 선행계획 수립 후 기계에 할당하는 방법[17]이고, 다른 하나는 기계할당 후 각 기계별 계획을 수립하는 방법[21]이다. 본 논문의 상황에서처럼 패밀리 셋업이 존재할 때, 패밀리간의 셋업 횟수를 줄이는 것이 허의 성능에 크게 영향을 미치는 경우에는 전자와 같은 방법을 따르는 것이 더 효율적이다. 따라서 본 논문에서는 전자와 같은 절차를 이용하여 전체 알고리즘을 선행계획 단계(pre-processing procedure)와 할당 단계(allocation procedure)로 구성하였다. 본 연구의 특징은 선행계획 단계에서 초기해를 구할 때, 매우 잘 계획된 일정계획을 수립함으로써 할당 단계에서는 해의 향상을 위한 별도의 절차가 필요치 않게 한다는 점이다. 즉, 선행계획 단계에서 패밀리 셋업과 납기, 우선순위 등의 중요한 요소들을 최대한 고려하여 작업순서를 결정하고, 이렇게 수립된 작업순서들의 처리순서는 할당 단계에서도 불가피한 경우를 제외하고는 변경되거나 재수립되지 않도록 한다.

선행계획 단계는 각 작업을 기계에 할당하기 전에 계획 대상 작업의 순서를 정해준다. 본 논문의 상황과 같이 패밀리 셋업이 존재하는 경우, 될 수 있는 한 같은 패밀리에 속하는 작업들을 하나의 패밀리로 묶어서 동일한 패밀리 셋업군 별로 계획을 수립하면 셋업 횟수를 줄일 수 있는 효과를 나타낼 것이다. 그러나 본 연구는 납기 관련 목적함수를 갖고 있기 때문에 전체 계획대상 작업을 단순히 패밀리 셋업군 별로 묶는다면 상대적으로 뒤쪽에 위치하는 패밀리 셋업군에 속하는 작업일수록 납기를 어기는 정도가 커지게 되어 좋지 못하다. 그러므로 본 논문에서는 앞에서 설명된 방식의 단점을 보완

하기 위하여 먼저 전체 계획 대상 작업을 비슷한 납기를 가지는 작업군으로 나누고, 이러한 각각의 작업군에 속하는 작업들에 대해서 패밀리군 별로 묶어 주는 방식으로 납기와 셋업 횟수를 동시에 줄여나갈 수 있도록 한다. 세부적으로 선행계획 단계에서는 납기의 우선순위에 따라 작업을 정렬하기 위해서 EDD(earliest due date) 규칙과 Lee와 Pinedo[15, 19]가 제안한 ATCS(apparent tardiness cost with setup) 규칙, 그리고 타부 탐색(tabu search) 기법을 적용하였다.

할당 단계에서는 선행계획 단계에서 수립된 계획 결과를 병렬기계에 할당하는 계획을 수립한다. 이러한 할당 단계에서는 할당 대상 작업의 납기와 패밀리 셋업 형태 등을 반영하여 목적함수를 최소화하는 할당 문제를 해결하기 위해 별도의 두 가지 할당 규칙을 제안하였다. 이에 대한 자세한 내용은 다음절에서 설명하기로 한다.

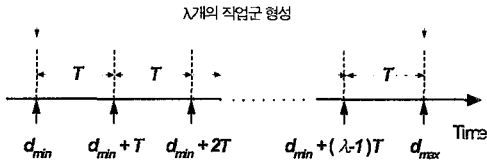
3.2.1 선행계획 단계

선행계획 단계에서는 각 작업을 병렬기계에 할당하기 전에 대상 작업의 순서를 정한다. 선행계획 단계의 첫 절차는 납기의 우선순위에 따라서 대상 작업들을 나열하는 것이다. 이를 위해 전체 계획 대상 작업을 앞서 언급했던 EDD 규칙에 의해 정렬한다. 이들을 비슷한 납기를 가지는 작업군으로 분류하기 위해서는 계획 대상 작업을 구분하는 기준이 필요한데 이를 위해 본 논문에서는 계획 시평(scheduling horizon)과 계획 구획 수(number of scheduling sections)를 정의한다. 계획 시평(T)은 계획 대상 작업의 납기 최대값(d_{max})과 최소값(d_{min})의 차이를 계획 구획 수(λ)로 나누어준 값이다. 이를 수식으로 나타내면 다음과 같다.

$$T = (d_{max} - d_{min}) / \lambda \quad (1)$$

이때, 납기 최소값을 기준으로 계획 시평만큼씩 증가시키고 해당 구간에 속하는 작업들을 하나의 작업군으로 정한다. 만약 분할한 계획 시평 가운데 작업이 존재하지 않은 구간이 없다면 작업군의 수

는 λ 와 일치하게 된다. 계획 시평 T 와 계획 구획 수 λ 의 관계를 표현하면 [그림 1]과 같다.



[그림 1] 계획 시평과 계획 구획 수의 관계

식 (1)과 [그림 1]에서 알 수 있듯이 λ 의 값이 커질수록 T 의 값은 작아지게 되어 보다 비슷한 납기를 가지는 작업들로 작업군이 형성되기 때문에 납기를 만족시킬 수 있는 경우는 많아질 수 있지만, 반대로 작업군의 수가 증가하게 되어 패밀리 셋업 발생 횟수를 증가시키는 효과를 나타낸다. 즉, λ 는 납기와 셋업 회수를 조정하는 역할을 하고 그 값에 따라 해의 성능이 좌우되므로 신중히 결정되어야 한다.

다음으로 ATCS 규칙을 이용하여 작업군내의 작업들에 대한 처리순서를 결정한다. 이때 ATCS 규칙은 작업군내의 작업들을 동일한 패밀리 셋업 형태를 갖는 패밀리 셋업군으로 묶어 주고, 동시에 패밀리 셋업군내의 작업들의 순서와 셋업군 사이의 순서를 조정하는 역할을 한다. ATCS 규칙은 순서의존적인 작업준비 시간(sequence-dependent setup time)을 고려하면서 납기 지연 가중치의 합을 최소화하는 것을 목적으로 하는 일정계획 수립에 사용되는 작업배정 규칙(dispatching rule)이다. ATCS 규칙에서 작업배정을 위해 사용하는 지표값(index)을 만드는 방안은 다음과 같이 표현된다.

$$I_j(t, i) = \frac{w_j}{p_j} \exp \left(- \frac{\max(d_j - p_j - t, 0)}{k_1 \bar{p}} \right) \exp \left(- \frac{s_{ij}}{k_2 \bar{s}} \right) \quad (2)$$

작업 i 가 t 시간에 특정 기계에서 마지막으로 처리되고 난 후 이를 기준으로 아직 일정계획이 수립되지 않은 나머지 작업들에 대하여 식 (2)와 같은

지표값을 계산한다. 이때 k_1 과 k_2 는 각각 납기와 작업 준비 시간에 관련된 scaling parameter를 나타낸다[15, 19]. 여기서 가장 큰 지표값을 갖는 작업이 작업 i 의 다음 계획 대상 작업으로 선정된다. 이러한 지표값을 계산하는 식 (2)의 우변은 세 개의 항목으로 이루어져 있다. 첫 번째는 작업별 가중치를 가공 시간으로 나누어주는 WSPT(weighted shortest processing time first) 규칙을 반영하는 항목으로 가공 시간이 짧은 작업들과 가중치가 큰 작업들을 우선적으로 계획하게 한다. 두 번째 항목은 MS(minimum slack) 규칙을 반영하는 것으로써 납기에 대한 여유가 적은 작업이 큰 지표값을 가지게 한다. 여기서 \bar{p} 는 일정계획 대상 작업의 평균 가공 시간을 나타낸다. 마지막 항목은 SST(shortest setup time) 규칙을 적용한 것으로 작업준비 시간이 상대적으로 짧은 작업이 먼저 계획될 수 있게 한다는 사실을 알 수 있다. 이때, \bar{s} 는 일정계획 대상 작업의 패밀리 셋업 시간을 모두 합한 값의 평균으로 정의하였으며, s_{ij} 항에는 동일 패밀리에 속하는 작업들을 하나의 패밀리 셋업군으로 묶어 주기 위하여 해당 작업의 패밀리 셋업 시간을 적용하였다.

이러한 ATCS 규칙이 패밀리 셋업이 존재하는 상황에 적용 될 경우에는 그 특성상 동일 패밀리에 속하는 작업들을 하나의 패밀리 셋업군으로 묶어 패밀리군 사이의 순서를 결정해주는 동시에 패밀리 셋업군 내의 작업들에 대한 계획을 수립해주는 역할을 하게 된다.

일반적으로 ATCS 규칙은 모든 후보 작업 중 가장 큰 지표값을 갖는 작업을 첫 번째 계획 대상 작업으로 선정하고 이 작업에 의존하여 패밀리군의 순서가 결정되는데 이 경우 ATCS 규칙에 의해 정해진 패밀리군의 순서가 반드시 전체 계획 대상 작업의 상황을 반영한다고 할 수는 없다. 예를 들어, 패밀리 A에 속하는 작업들은 그 중 하나가 모든 후보 작업들 중 가장 큰 지표값을 갖고 나머지 작업은 상대적으로 작은 지표값을 갖는데 반해, 패밀리 B에 속하는 작업들은 평균적으로 높은 지표값을 갖는다고 하자. 이러한 경우에 ATCS 규칙을 적용한

다면 가장 큰 지표값을 갖는 패밀리 A에 속하는 작업을 첫 번째 계획 대상 작업으로 선정하고 패밀리 A에 속하는 나머지 작업들이 이후의 계획 대상 작업으로 결정될 것이다. 그러나 모든 계획 대상 작업의 상황을 고려한다면 평균적으로 지표값이 크게 나타난 패밀리 B에 속하는 작업부터 계획하는 것이 더 효과적일 것이다.

본 연구에서는 이와 같은 ATCS 규칙의 단점을 보완하여 패밀리군의 순서 조정을 위해서 타부 탐색(tabu search) 기법을 추가로 사용하였다. Glover [7,8]에 의해 체계화된 타부 탐색은 일반적으로 복합 최적화(combinatorial optimization)와 일정계획 수립 문제 등의 해결에 다양하게 사용되고 있다. 타부 탐색의 특성을 이용하여 동일 패밀리 셋업군 내에 포함된 작업들 사이의 순서와 패밀리 셋업군들 사이의 순서를 다시 조정하게 된다. 타부 탐색의 해법 절차는 우선 임의의 초기해(initial solution)로부터 이웃해(neighborhood solution)를 탐색해 나아가는 과정으로써, 이미 탐색이 완료된 해의 경우 타부 목록(tabu list)에 넣어 재차 탐색하지 않도록 하여 지역해(local solution)에서 벗어나게 하는 방법을 말한다. 일반적으로 타부 탐색의 성능에 영향을 미치는 요소로서는 이웃해 생성 방법, 초기해 성능, 타부 목록의 크기, 종료 조건(termination condition) 등이다[9, 13]. 본 연구에서의 초기해는 ATCS 규칙에 의해서 일차적으로 편성된 패밀리 셋업군 내의 작업이 이에 해당된다. 타부 탐색에 의하여 이러한 초기해에 대한 성능 개선이 모두 마쳐지면 이차적으로 패밀리 셋업군 사이의 순서가 초기해가 되고 이를 다시 향상시키는 과정을 거치게 된다. 첫 번째 과정은 ATCS 규칙이 단순한 작업배정 규칙(dispatching rule)으로써 최적해를 보장해 주지 못하므로 패밀리 셋업군 내의 작업의 순서를 조정해 줌으로써 보다 개선된 해를 얻게 하기 위함이고, 두 번째 과정은 앞서 언급했던 ATCS 규칙의 단점을 보완하여 패밀리 셋업군 사이의 순서를 조정해 주기 위한 것이다.

본 연구에서의 이웃해의 생성은 기준 작업에

따라 달리 정의된다. 기준 작업은 작업의 위치에 따라 변경되는데, 먼저 대상 작업 중 첫 번째 작업을 기준 작업으로 선정하고 이러한 기준 작업(이 경우 첫 번째 작업)과 교환(swap)하여 가장 좋은 목적함수 값의 향상을 가져올 수 있는 작업을 후보 작업으로 선택하고 이 두 작업의 위치를 바꾼다. 다음으로 두 번째 작업이 기준 작업으로 선정되고 교환할 대상 작업을 찾아가는 과정을 반복하게 된다.

지금까지 설명한 선행계획 단계의 절차와 이를 표현하기 위한 추가적인 용어를 정리하면 다음과 같다.

<용어 정리>

- c : 작업군의 수, $c=1,2,\dots,\lambda$
- $L(c)$: c 번째 작업군에 속한 작업, $c=1,2,\dots,\lambda$
- $L_{ATCS}(c)$: ATCS 규칙을 적용한 $L(c)$, $c=1,2,\dots,\lambda$
- $L_{TS}(c)$: 타부 탐색을 적용한 $L_{ATCS}(c)$, $c=1,2,\dots,\lambda$

<선행계획 단계 절차>

Step 0 : 초기화

작업들을 납기 우선순위에 따라 정렬하고 계획 구획 수 λ 결정.

Step 1 : $L(c)$ 결정, $c=1,2,\dots,\lambda$

Step 2 : $L(c)$ 에 ATCS 규칙을 적용하여

$L_{ATCS}(c)$ 생성, $c=1,2,\dots,\lambda$

Step 3 : $L_{ATCS}(c)$ 에 타부 탐색을 적용하여

$L_{TS}(c)$ 생성, $c=1,2,\dots,\lambda$

Step 4 : 만약 $c=\lambda$ 이면 종료. 그렇지 않다면

$c=c+1$ 로 설정하고 Step 2로 이동.

Step 0은 초기화 단계로 전체 계획 시평(scheduling horizon) 내에 속한 모든 작업들을 납기의 순서대로 정렬한다. Step 1에서는 정렬된 작업들을 계획 구획 수에 따라 λ 개의 작업군($L(c)$)으로 나눈다. 이렇게 나누어진 작업군들은 서로 비슷한 납기

를 가지는 작업들로 구성되어지게 된다. Step 2에서는 ATCS 규칙을 이용하여 $L(c)$ 에 속한 작업들의 일정계획을 수립하고, Step 3에서 타부 탐색을 통해서 이를 개선시킨다. Step 4에서는 전체 계획 대상 작업에 대한 선행계획이 모두 끝났는지를 체크한다. 즉, 현재까지 계획된 작업군의 수인 c 가 계획 구획 수 λ 와 같다면 선행계획 단계를 종료하고, 그렇지 않다면 c 를 1만큼 증가시킨 후 Step 2부터의 과정을 반복한다.

3.2.2 할당 단계

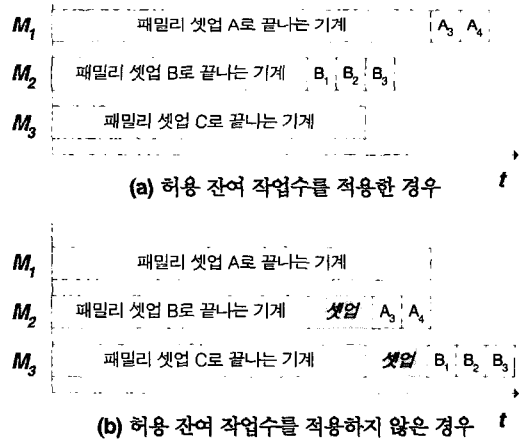
할당 단계에서는 선행계획 단계에서 수립된 일정계획을 바탕으로 작업들을 동일한 병렬기계에 할당한다. 할당 단계에서는 할당 대상 작업의 납기와 패밀리 셋업 형태 등을 반영하여 목적함수인 납기지연 가중치의 합을 최소화 하는 할당 계획을 수립한다. 할당의 기본 원칙은 패밀리 셋업 시간이 가공 시간에 비해 상대적으로 월등히 크므로 될 수 있는 한 같은 패밀리에 속하는 작업은 연속해서 동일한 패밀리로 끝나는 기계에 할당하여 패밀리 셋업 횟수를 줄이는 것이다. 그러나 본 연구는 납기관련 목적함수를 갖고 있기 때문에 납기도 함께 고려되어야 한다. 그러므로 언제나 이러한 원칙이 적용될 수 있는 것은 아니다. 예를 들어, 현재 할당되어야 하는 작업과 동일한 패밀리 셋업 형태로 끝난 기계가 존재하고 이 작업이 해당 기계에 할당될 경우 납기를 만족한다면 패밀리 셋업도 발생하지 않고 납기 또한 만족시킬 수 있으므로 당연히 해당 기계에 할당하는 것이 바람직하다. 그러나 어떠한 경우에는 납기를 어느 정도 어기는 경우가 발생하더라도 패밀리 셋업의 형태를 먼저 고려하여 동일한 패밀리 셋업 형태로 끝나는 기계에 계속해서 작업을 할당하는 것이 전체적인 일정계획의 측면에서 효과적일 수 있을 것이다. 그러므로 대상 작업을 할당하는데 있어서 보다 세부적이고 명확한 기준이 필요한데 이를 위해서 본 연구에서는 납기지연 한계값(threshold value, α)과 허용 잔여 작업수(look-ahead parameter, β)를 정의하고 이를 바탕으로 할

당규칙을 적용한다.

납기지연 한계값(threshold value)은 할당 대상 작업과 동일한 패밀리 셋업 형태로 끝난 기계가 존재할 경우, 만약 이 작업이 해당 기계에 할당되었을 때 납기를 만족시키지 못하더라도 납기지연의 정도(tardiness)가 미리 지정된 한계값 보다 작다면 해당 기계에 할당할 수 있도록 하는 지표이다. 이는 어느 일정 한도 이내로 납기를 어기더라도 셋업을 발생시키지 않는 것이 다른 패밀리 셋업 형태로 끝나는 기계에 할당되어 셋업을 발생시키는 것보다 전체 일정계획의 측면에서 효과적인 수 있기 때문이다. 본 연구에서는 전체 패밀리 셋업 시간 중의 최대값을 납기지연 한계값으로 설정하였다.

$$L_{TS} = \{A_3, A_4, B_1, B_2, B_3, \dots\}$$

Look-ahead parameter = 2



[그림 2] 허용 잔여 작업수의 예

허용 잔여 작업수(look-ahead parameter)는 현재 할당 대상 작업을 기준으로 뒤에 오는 몇 개의 작업까지를 연속하여 동일한 패밀리 셋업으로 끝나는 기계에 할당할 것인가를 나타내는 모수이다. 이는 다음에 위치할 패밀리군이 할당될 때 발생하는 셋업의 횟수를 줄이기 위해, 현재 할당 대상 작업이 포함된 패밀리군의 잔여 작업수가 look-ahead parameter 값 이하일 경우 납기와 관계없이 동일한 패밀리로 끝나는 기계에 강제적으로 할당하도록 하

는 것을 의미한다. [그림 2]는 허용 잔여 작업수 (look-ahead parameter)를 적용했을 경우와 그렇지 않았을 경우의 차이를 간트 차트(Gantt chart)를 통해 비교하여 설명하고 있다.

[그림 2]에서 현재 할당 대상 작업은 A_3 이고 이와 동일한 패밀리 셋업 형태로 끝난 기계 M_1 또한 존재한다. 그러나 M_1 에는 이미 많은 작업이 할당되어 있어 A_3 가 할당된다면 납기를 어기는 정도가 납기 지연 한계값 이상이 된다고 가정하자. (b)는 이러한 상황에서 동일 잔여 작업수를 적용하지 않은 경우로, A_3 와 A_4 는 대상 기계 중 가공 완료시간이 가장 빠른 M_2 에 할당되고 다음에 위치한 계획 대상 작업 B_1, B_2, B_3 등은 동일한 패밀리 셋업 형태로 끝나는 기계가 존재하지 않으므로 M_3 에 할당되었다. (a)의 경우 허용 잔여 작업수를 적용하였는데, 현재 허용 잔여 작업수가 2로 설정되어 있으므로 A_3 와 A_4 2개의 작업은 납기를 어기더라도 동일한 패밀리로 끝난 M_1 에 할당되고, 이로 인해서 B_1, B_2, B_3 의 작업 또한 동일한 패밀리로 끝난 기계 M_2 에 할당될 수 있게 된다. (a)의 경우 비록 A_3 와 A_4 를 할당하는데 있어서 어느 정도의 납기 지연이 발생하였지만 패밀리 셋업이 한 번도 발생되지 않아 두 번의 패밀리 셋업이 발생된 (b)의 경우보다 효과적이라 할 수 있다.

지금까지 설명한 할당 단계의 절차와 이를 설명하기 위한 추가적인 용어를 정리하면 다음과 같다.

<용어 정리>

- M : 전체 계획대상 기계의 집합, $m \in M$
- J : 전체 계획대상 작업의 집합, $j \in J$
- f_j^1 : 작업 j 의 패밀리 셋업 형태, $j \in J$
- C_m : 기계 m 에 계획된 마지막 작업의 완료시간, $m \in M$
- C_{jm} : 작업 j 가 기계 m 에 할당되었을 경우의 완료시간, $m \in M$
- f_m^2 : 기계 m 에 마지막으로 할당된 작업의 패밀리 셋업 형태, $m \in M$

<할당 단계 절차>

Step 0 : 초기화

$c=1$ 로 설정한 후 $L(c)$ 에 속하는 첫 작업 j 를 기계 1에 할당.

$$L(c) = L(c) \setminus j.$$

Step 1 : $L(c)$ 의 첫 작업 j 선택.

만약 $f_j^1 = f_m^2$ 인 기계가 존재하고 이중 가장 작은 C_{jm} 을 갖는 기계 m 에 대해 $C_{jm} \leq d_j$ 이면 작업 j 를 기계 m 에 할당하고 Step 4로 이동.

$C_{jm} > d_j$ 이면 Step 2로 이동.

$f_j^1 = f_m^2$ 인 기계가 존재하지 않으면 Step 3으로 이동.

Step 2 : 작업 j 가 납기 지연이 발생했을 경우 α (즉, $C_{jm} \leq d_j + \alpha$) 및 β 검사.

만약 작업 j 가 α 또는 β 중에서 어느 하나라도 만족한다면 기계 m 에 할당하고 Step 4로 이동.

그렇지 않다면 Step 3으로 이동.

Step 3 : 작업 j 를 모든 대상 기계 중에서 가장 작은 C_{jm} 을 갖는 기계 m 에 할당하고 Step 4로 이동.

Step 4 : $L(c) = L(c) \setminus j$.

만약 $L(c) = \emptyset$ 이면 Step 5로 이동.

그렇지 않다면 Step 1로 이동.

Step 5 : 만약 $c = \lambda$ 이면 종료.

그렇지 않다면 $c = c + 1$ 이고 Step 1로 이동.

Step 0은 초기화 단계로 첫 번째 계획대상 작업군 ($L(1)$)에 속하는 작업 가운데 맨 처음에 위치하는 작업을 첫 번째 기계에 할당하고 이를 계획 대상 작업군에서 제외시킨다. Step 1에서는 $L(c)$ 의 첫 번째 작업 j 를 선택하여 동일한 패밀리 셋업 형태로 끝나는 기계의 존재 여부를 확인하고, 만약 그러한 기계가 존재하는 경우 작업 j 가 해당 기계에 할당되었을 때 납기를 만족시키는지 판단한다. 만약 위의 조건을 만족한다면 작업 j 를 해당 기계인 m 에 할당하고, 그렇지 않을 경우 또 다른 확인

과정을 거치기 위해 Step 2로 간다. 그리고 작업 j 와 동일한 패밀리 셋업 형태로 끝나는 기계가 존재하지 않는다면 Step 3으로 간다. Step 2에서는 Step 1에서 처리하지 못한 작업 j 가 납기지연 한계값과 허용 잔여 작업수를 만족시키는지를 검사한다. 즉, 작업 j 가 기계 m 에 할당될 경우 비록 납기를 만족시키지 못하지만 동일한 패밀리 셋업 형태를 가지고 있으므로, 납기지연 한계값과 허용 잔여 작업수를 검사하여 이 두 가지 중에서 어느 하나의 조건이라도 만족시킬 경우 해당 기계 m 에 할당한다. 이렇게 작업 j 가 납기를 만족시키지 못함에도 불구하고 기계 m 에 강제로 할당하는 것은 앞서 설명한 것과 마찬가지로 납기를 조금 어기더라도 이후에 발생하게 될 추가적인 패밀리 셋업의 횟수를 줄여 주기 위함이다. 작업 j 가 납기지연 한계값이나 허용 잔여 작업수 중에 어느 하나의 규칙도 만족시키지 못하거나, 동일한 패밀리 셋업 형태로 끝나는 기계가 존재하지 않는 경우에는 Step 3에서처럼 작업 완료시간이 가장 빠른 기계에 할당한다. 만약 가장 빠른 기계가 여러 개 존재한다면 임의의 기계에 할당하여도 무방하다. Step 4는 할당된 작업을 계획 대상 작업군에서 제외시키고, 이후에 만약 계획 대상 작업군내의 모든 작업에 대하여 할당이 완료되었다면 Step 5로, 그렇지 않다면 Step 1로 돌아가 앞서 설명한 할당 절차를 반복한다. Step 5는 전체 계획 대상 작업군에 대한 할당이 끝났는지를 체크한다. 즉, 현재까지 계획된 작업군의 수가 계획 구획수와 같다면 할당 단계를 종료하고, 그렇지 않았다면 다음 계획 대상 작업군을 선정하고 할당 절차를 반복한다.

4. 실험 계획 및 결과

4.1 수치 예제

본 수치 예제는 3개의 기계, 15개의 작업과 4개의 패밀리 셋업 형태를 갖는 문제를 다룬다. 다음 <표 1>은 이 문제에 대한 기초 데이터로서 대상 작업의

패밀리 타입, 납기, 가중치 및 가공 시간을 나타낸다. 각 작업들의 패밀리 형태(A, B, C, D)별 셋업 시간은 각각 15, 18, 16, 20이다. 계획 구획 수(λ)는 2이며, 납기지연 한계값(α)과 허용 잔여 작업수(β)는 각각 20과 2로 정하였다.

먼저 선행계획 단계로서 초기화 절차를 수행하면, 모든 작업들은 납기가 빠른 순서대로 정렬된다. 즉, 작업들은 8, 3, 15, 12, 1, 9, 5, 2, 6, 13, 7, 10, 4, 11, 14의 순으로 정렬된다. 그리고 대상 작업의 납기를 근거로 하여 d_{min} 은 14, d_{max} 는 40으로 각각 정해진다. 따라서 계획 시평 $T = (d_{max} - d_{min}) / \lambda = 13$ 이 된다. Step 1에서 첫 번째 및 두 번째 작업군은 납기가 각각 $14 \leq d_j \leq 27$ 과 $27 < d_j \leq 40$ 에 속하는 작업으로 결정된다. 즉, $L(1) = \{8, 3, 15, 12, 1, 9, 5, 2, 6\}$ 이고 $L(2) = \{13, 7, 10, 4, 11, 14\}$ 가 된다.

<표 1> 예제 데이터

작업	패밀리 타입	납기	가중치	가공 시간
1	A	20	3	6
2	C	25	3	8
3	B	15	5	7
4	B	30	8	8
5	C	23	4	5
6	C	26	3	7
7	D	32	6	6
8	A	14	8	9
9	B	21	5	6
10	D	29	5	7
11	B	35	8	10
12	C	19	1	8
13	D	29	8	8
14	A	40	3	9
15	D	15	4	9

다음으로 Step 2에서 $L(1)$ 에 ATCS 규칙을 적용하여 $L_{ATCS}(1)$ 을 구하면 $L_{ATCS}(1) = \{8(A), 1(A), 3(B), 9(B), 12(C), 5(C), 2(C), 6(C), 15(D)\}$ 가 된다. 마찬가지로 $L_{ATCS}(2) = \{13(D), 10(D), 7(D), 14(A), 4(B), 11(B)\}$ 를 구할 수 있다. Step 3에서는

다시 $L_{ATCS}(1)$ 과 $L_{ATCS}(2)$ 에 타부 탐색을 적용하여 동일한 패밀리 셋업을 갖는 작업 집합 내의 작업들의 순서와 패밀리군 사이의 우선순위를 재조정하면 다음과 같다.

$$L_{TS}(1) = \{3(B), 9(B), 8(A), 1(A), 5(C), 6(C), 2(C), 12(C), 15(D)\}$$

$$L_{TS}(2) = \{13(D), 7(D), 10(D), 4(B), 11(B), 14(A)\}$$

선행계획 단계가 완료되었으면 할당 단계를 수행하는데, 할당 단계의 과정은 다음과 같다.

Step 0 : $L_{TS}(1)$ 의 첫 작업인 $j=3$ 을 첫 번째 기계 $m=1$ 에 할당. $L_{TS}(1) = \{9(B), 8(A), 1(A), 5(C), 6(C), 2(C), 12(C), 15(D)\}$ 로 변경.

Step 1 : $L_{TS}(1)$ 의 첫 작업인 $j=9$ 와 동일한 패밀리 셋업(B) 형태로 끝나는 기계 $m=1$ 이 존재하고, $C_{9,1} > d_9$.

Step 2 : $C_{9,1} - d_9 = 10 < \alpha (= 20)$ 이므로 작업 $j=9$ 를 기계 $m=1$ 에 할당.

Step 4 : $L_{TS}(1) = \{8(A), 1(A), 5(C), 6(C), 2(C), 12(C), 15(D)\}$.

Step 1 : $L_{TS}(1)$ 의 첫 작업인 $j=8$ 와 동일한 패밀리 셋업 형태로 끝나는 기계가 존재하지 않는다.

Step 3 : 작업 $j=8$ 을 가공완료 시간이 가장 빠른 기계인 $m=2$ 에 할당.

Step 4 : $L_{TS}(1) = \{1(A), 5(C), 6(C), 2(C), 12(C), 15(D)\}$.

Step 1 : $L_{TS}(1)$ 의 첫 작업인 $j=1$ 과 동일한 패밀리 셋업(A) 형태로 끝나는 기계 $m=1$ 이 존재하고, $C_{1,2} > d_1$.

Step 2 : $C_{1,2} - d_1 = 10 < \alpha (= 20)$ 이므로 작업 $j=9$ 을 기계 $m=2$ 에 할당.

Step 4 : $L_{TS}(1) = \{5(C), 6(C), 2(C), 12(C), 15(D)\}$

로 변경.

Step 1 : $L_{TS}(1)$ 의 첫 작업인 $j=5$ 와 동일한 패밀리 셋업 형태(C)로 끝나는 기계가 존재하지 않는다.

Step 3 : 작업 $j=5$ 를 가공완료 시간이 가장 빠른 기계인 $m=3$ 에 할당.

Step 4 : $L_{TS}(1) = \{6(C), 2(C), 12(C), 15(D)\}$.

Step 1 : $L_{TS}(1)$ 의 첫 작업인 $j=6$ 과 동일한 패밀리 셋업(C) 형태로 끝나는 기계 $m=3$ 이 존재하고, $C_{6,3} > d_6$.

Step 3 : $C_{6,3} - d_6 = 2 < \alpha (= 20)$ 이므로 작업 $j=6$ 을 기계 $m=3$ 에 할당.

Step 4 : $L_{TS}(1) = \{2(C), 12(C), 15(D)\}$.

Step 1 : $L_{TS}(1)$ 의 첫 작업인 $j=2$ 와 동일한 패밀리 셋업(C) 형태로 끝나는 기계 $m=3$ 이 존재하고, $C_{2,3} > d_2$.

Step 2 : $C_{2,3} - d_2 = 11 < \alpha (= 20)$ 이므로 작업 $j=2$ 를 기계 $m=3$ 에 할당.

Step 4 : $L_{TS}(1) = \{12(C), 15(D)\}$.

Step 1 : $L_{TS}(1)$ 의 첫 작업인 $j=12$ 와 동일한 패밀리 셋업(C) 형태로 끝나는 기계 $m=3$ 이 존재하고, $C_{12,3} > d_{12}$.

Step 2 : $C_{12,3} - d_{12} = 25 > \alpha (= 20)$ 이지만, $j=12$ 는 패밀리 셋업 형태가 C인 마지막 작업이므로 남은 작업수가 1로 $\beta=2$ 보다 작으므로 작업 $j=12$ 를 기계 $m=3$ 에 할당.

Step 4 : $L_{TS}(1) = \{15(D)\}$.

Step 1 : $L_{TS}(1)$ 의 첫 작업인 $j=15$ 와 동일한 패밀리 셋업 형태(D)로 끝나는 기계가 존재하지 않는다.

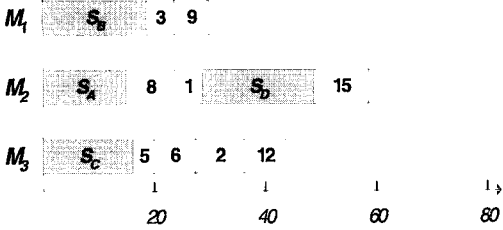
Step 3 : 작업 $j=15$ 를 가공완료 시간이 가장 빠른 기계인 $m=2$ 에 할당.

Step 4 : $L_{TS}(1) = \{\emptyset\}$.

Step 5 : $c (= 1) < \lambda (= 2)$ 이므로 $c=2$ 로 변경하고 Step 1로 이동.

이와 같은 과정을 거쳐 완성된 첫 번째 작업군 ($c=1$)에 대한 일정계획 결과는 [그림 3]과 같다. 또한 이러한 과정을 전체 계획 대상 작업에 대해 반복적으로 수행하면 최종적으로 [그림 4]와 같은 일정계획 결과가 얻어진다.

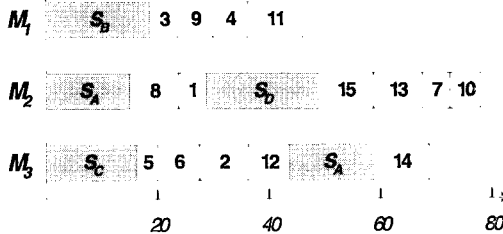
$$L_{IS}(1) = \{3(B), 9(B), 8(A), 1(A), 5(C), 6(C), 2(C), 12(C), 15(D)\}$$



[그림 3] 첫 번째 작업군에 대한 일정 계획 결과

$$L_{IS}(1) = \{3(B), 9(B), 8(A), 1(A), 5(C), 6(C), 2(C), 12(C), 15(D)\}$$

$$L_{IS}(2) = \{13(D), 7(D), 10(D), 4(B), 11(B), 14(A)\}$$



[그림 4] 전체 계획 대상 작업에 대한 일정계획 결과

4.2 실험 설계 및 결과 분석

본 논문에서 제안된 알고리즘의 성능을 평가하기 위해 기존에 알려진 알고리즘을 비교대상으로 하여 다양한 실험을 실시하여 분석하였다. 앞서 기존 연구에서 언급하였듯이 본 연구와 동일한 상황에 대한 알고리즘은 존재하지 않으므로 이와 가장 유사하다고 할 수 있는 Ovacik & Uzsoy[17, 18]가 제안한 RHP(rolling horizon procedure)와 Lee & Pinedo[15]가 제안한 ATCS 알고리즘을 비교 대안으로 정하였다. RHP는 순서 의존적인 작업준비 시간을 갖는 병렬기계 문제를 일련의 단일기계 문제로 분

할(decomposition)하고 각각의 단일기계 문제에 대해 분지 한계법을 이용하여 납기 지연 시간의 최대 값(L_{max})을 최소화하는 방법이고, ATCS 알고리즘은 기존의 단일기계 문제에 적용했던 ATCS 알고리즘을 병렬기계 문제로 확장한 것으로 순서 의존적인 작업준비 시간을 갖는 병렬기계 문제에 대한 납기 지연 가중치의 합을 최소화하는 방법이다.

실험에 필요한 데이터를 얻기 위해 작업의 가공 시간, 가중치, 패밀리 셋업 시간, 납기는 난수를 발생시켜 각각 (1, 300), (1, 10), (300, 500), (500, 1000)의 범위를 갖는 균일 분포(uniform distribution)를 이루도록 생성하였다. 또한 계획 구획 수(λ)는 2로 고정시켰고, 납기 지연 한계값(α)은 실험데이터 중에서 가장 큰 패밀리 셋업 시간으로 설정하였으며 허용 잔여 작업 수(β)는 2로 고정한다. 문제의 유형은 작업 수(n) = (30, 50, 100), 기계 수(m) = (3, 5, 7, 10), 패밀리 셋업 수(F) = (4, 6, 8)의 조합으로 총 36가지로 구성하였으며, 각각의 문제들은 20번의 반복 실험을 실시한 후 그 결과의 평균값을 제시하였다.

<표 2>에서는 RHP, ATCS와 제안된 알고리즘의 수행 결과에 대한 비교값을 나타내었다. <표 2>에서 알 수 있듯이 본 논문에서 제안한 알고리즘이 기존의 비교대상 알고리즘보다 좋은 결과를 얻고 있다는 것을 보여준다. 이는 본 논문의 알고리즘이 선행계획과 할당 단계를 통해 납기와 패밀리 셋업의 회수를 동시에 고려하는 효과적인 계획수립을 하는 반면에, RHP는 계획되어진 작업군을 한꺼번에 하나의 기계에 할당하기 때문에 비효율적이라 할 수 있다. ATCS의 경우 패밀리 타입과 기계수가 적을 경우 제안된 알고리즘과 근사한 비교적 좋은 결과값을 보이지만 이 수가 증가할수록 좋지 못한 결과를 나타내었다.

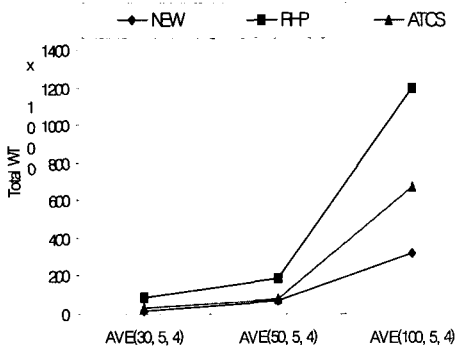
[그림 5]는 $m = 5, F = 4$ 로 고정시키고 작업 수의 변화에 따른 각 알고리즘의 성능 변화를 나타낸 것이다. 여기서 $AVE(n, m, F)$ 는 n 개의 작업, m 개의 기계, 그리고 F 개의 패밀리 셋업의 형태를 갖는 문제를 의미한다. [그림 5]에서 작업수가 증가할

〈표 2〉 성능 비교 테이블

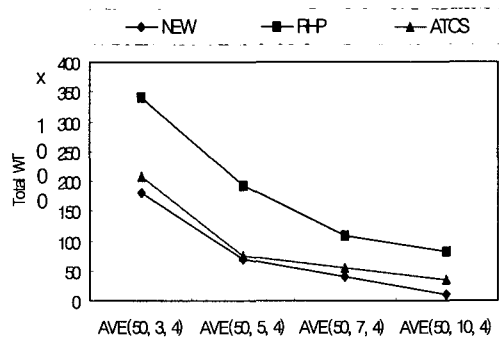
n	m	F = 4			F = 6			F = 8		
		NEW	RHP	ATCS	NEW	RHP	ATCS	NEW	RHP	ATCS
30	3	67,613	167,609	67,684	145,670	225,476	213,411	101,845	213,823	200,131
	5	18,477	8,9843	28,511	30,910	125,033	146,826	38,795	133,590	101,192
	7	10,771	55,315	19,607	16,288	83,425	84,835	20,174	82,921	58,090
	10	8,477	25,379	16,599	11,42	49,705	35,162	12,398	50,640	43,613
50	3	17,9610	340,673	207,863	194,274	476,547	347,677	251,847	508,111	325,016
	5	70,189	191,364	74,895	81,179	264,883	163,551	124,712	297,087	241,023
	7	39,961	109,690	53,511	40,056	172,312	147,466	57,736	214,337	175,212
	10	9,345	82,599	34,792	12,840	122,968	98,566	21,953	117,232	103,354
100	3	948,963	2,102,915	1,069,266	1,298,167	2,239,811	1,936,631	1,406,763	2,637,634	2,292,707
	5	322,424	1,200,069	673,700	484,508	1,285,489	827,227	495,633	1,531,391	1,309,291
	7	271,918	825,372	507,624	358,192	820,991	610,789	417,761	1,014,218	764,492
	10	181,799	559,188	325,128	235,988	532,766	522,793	280,936	635,094	535,677

수록 각 알고리즘의 수행도의 차이도 증가하는데 특히 작업수가 50이상일 경우 그 차이는 훨씬 커지는 것을 알 수 있다. 예를 들어, 작업수가 100일 때 RHP는 제안된 알고리즘보다 3배 이상이나 큰 결과 값을 보였다. 이러한 결과는 RHP의 경우 분지 한계법(branch and bound)을 이용하기 때문으로, Sc-hutten & Leussink[20]의 연구 결과에 의하면 일반적으로 분지 한계법의 경우 20개 이내의 작업 수에서는 우수한 결과를 얻는 반면에 그 이상의 경우 수행속도나 해의 성능면에서 좋지 못한 결과를 얻게 된다.

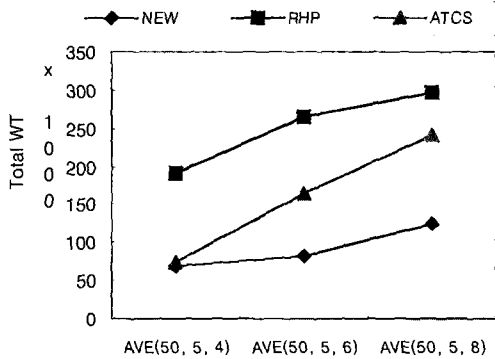
[그림 6]은 $n = 50, F = 4$ 로 고정시키고 기계 수의 변화에 따른 각 알고리즘의 성능 변화를 나타낸 것이다. 세 알고리즘 모두가 기계 수의 증가에 따라 목적함수의 값이 향상됨을 알 수 있다. 일반적으로 패밀리 셋업이 존재하는 경우의 일정계획 수립 시에는 패밀리 셋업의 수보다 가용한 기계의 수가 많을 경우에는 비교적 효율적으로 패밀리 셋업의 형태별로 작업을 할당할 수 있다. 그러나 가용 기계의 수가 패밀리 셋업 수보다 적을 경우에는 할당되어야 할 작업들이 보다 빈번하게 셋업의 형태를 변경하여야 하기 때문에 결과적으로 전체 수행도는 열등해진다.



[그림 5] 작업 수의 변화에 따른 수행도 비교



[그림 6] 기계 수의 변화에 따른 수행도 비교



[그림 7] 패밀리 셋업 수의 변화에 따른 수행도 비교

[그림 7]은 $n = 50$, $m = 5$ 로 고정시키고 패밀리 셋업 수의 변화에 따른 각 알고리즘의 성능 변화를 나타낸 것이다. [그림 7]에서 제안된 알고리즘은 나머지 알고리즘보다 모든 경우에 우월한 수행도를 보이고 있다. 특히 제안된 알고리즘은 다른 알고리즘에 비해 패밀리 셋업수의 증가에 따른 목적함수의 증가치가 상대적으로 완만함을 알 수 있으며, 이로 인해 제안된 알고리즘은 패밀리 셋업이 존재하는 환경에서 우수한 결과를 나타낸다고 할 수 있겠다.

4.3 계획 구획 수에 따른 수행도 분석

선행계획 단계에서 최초 납기 우선순위에 따라 정렬된 작업들의 계획 구간을 λ 에 따라 세분화하였다. λ 는 계획 구간을 작은 구간으로 세분화하여 납기와 패밀리 셋업 횟수간의 상충되는 관계를 조정해주는 역할과 문제해결의 편의성과 계산의 신속성을 제공한다.

<표 3>은 $n = 70$, $m = 6$, $F = 5$ 인 실험환경에서 α 와 β 를 변화시켜가면서, 15번의 실험을 통하여 얻어지는 최적의 λ 값과 각 기계(M_1, \dots, M_6)에서의 납기지연 가중치의 합 및 이들의 전체 총합을 나타내고 있다. 표에서 최적의 λ 값은 일정하지 않으며, 이러한 현상은 실험을 위한 데이터 생성 시 각각의 문제별 데이터가 동일하지 않음으로 인해 발생하는 결과이다. 그러므로 일정계획의 대상이 되는 작업의 환경과 특성에 따라서 α 와 β 및 λ 의 값을 적

절히 조절해 주면서 제안된 알고리즘을 적용한다면, 보다 우수한 결과를 얻을 수 있으리라 판단된다.

<표 3> 각 실험별 최적의 λ 값의 변화

M_1	M_2	M_3	M_4	M_5	M_6	Total	Best λ
30,364	15,415	22,073	25,836	6,686	7,950	108,324	4
13,310	13,154	11,707	11,097	3,152	8,195	60,615	5
11,618	3,175	1,196	14,777	27,137	796	58,699	2
35,160	8,109	16,853	6,080	38,968	20,753	125,923	3
27,060	19,260	25,692	13,640	7,194	18,634	111,480	3
2,825	7,280	8,768	36,158	43,807	6,077	104,915	2
2,634	10,936	7,736	12,715	7,709	1,531	43,261	4
15,240	11,215	5,990	10,388	3,752	22,395	68,980	2
16,455	27,459	20,685	37,693	12,705	22,951	137,948	2
3,603	5,248	18,935	5,930	11,192	22,888	67,796	3
13,660	14,371	3,864	4,554	24,911	18,797	80,157	2
3,398	17,599	11,588	5,619	23,569	19,060	80,833	4
20,988	26,976	31,400	12,470	11,550	5,529	108,913	2
9,042	11,466	5,303	27,113	19,698	9,091	81,713	3
28,150	7,961	20,838	19,513	17,877	19,572	113,911	4

5. 결론

본 연구에서는 패밀리 셋업이 존재하는 동일 병렬기계 문제에 대한 납기지연 가중치의 합을 최소화하기 위한 발견적 기법(heuristic)을 제안하였다. 제안된 발견적 기법은 크게 선행계획 단계와 할당 단계의 두 단계로 구성되는데, 선행계획 단계에서는 먼저 대상 작업을 납기의 우선순위에 따라 정렬하고 계획 구획 수(number of scheduling sections)를 사용하여 비슷한 납기를 가지는 작업군으로 분류하였다. 그리고 ATCS 알고리즘을 사용하여 동일 패밀리 셋업을 갖는 작업별로 패밀리군을 형성하였으며, 타부 탐색으로 세부 일정계획을 발전시켰다. 할당 단계에서는 이전 단계에서 수립된 선행계획 결과를 납기와 패밀리 셋업 횟수 등을 고려하여 각 기계에 할당할 수 있도록 납기지연 한계값(threshold value)과 허용 잔여 작업수(look-ahead parameter)라는 모수를 적용하였다.

본 연구에서는 제안된 알고리즘의 객관적인 성능

평가를 위해 RHP와 ATCS 알고리즘을 이용하여 비교 실험을 수행하였고 다양한 실험을 통해 제안된 알고리즘 우수성을 입증하였다. 추후 연구로는 작업에 대한 공정 가능 시간(release time)이 존재하는 경우 이를 고려할 수 있는 방법과 작업을 기계에 할당하는 과정에서 작업 분할(job split)을 고려할 수 있는 방법 등에 대한 연구가 필요하다. 또한 실제 현장에서 적용이 가능하도록 동적인 상황에서 계획수립(dynamic scheduling)이 가능하도록 하는 연구가 이루어져야 한다.

참 고 문 헌

- [1] Azizoglu, M. and S. Webster, "Scheduling Parallel Machines to Minimize Weighted Flow-time with Family Set-up Times," *International Journal of Production Research*, Vol.41, No.6 (2003), pp.1199-1215.
- [2] Barnes, J.W. and J.W. Brennan, "An Improved Algorithm for Scheduling Jobs on Identical Machines," *IIE Transactions*, Vol.9, No.1(1977), pp.25-31.
- [3] Bitran, G.R. and S.M. Gilbert, "Sequencing Production on Parallel Machines with Two Magnitudes of Sequence-Dependent Setup Cost," *Journal of Manufacturing and Operations Management*, Vol.3(1990), pp.190-206.
- [4] Cheng, T.C.E. and C.C.S. Sin, "A State-of-the-Art Review of Parallel-Machine Scheduling Research," *European Journal of Operational Research*, Vol.47, No.3(1990), pp.271-292.
- [5] Dogramaci, A., "Production Scheduling of Independent Jobs on Parallel Identical Processors," *International Journal of Production Research*, Vol.22, No.4(1984), pp.535-548.
- [6] Elmagraby, S.E. and S. Park, "Scheduling Jobs on a Number of Identical Machines," *IIE Transactions*, Vol.6, No.1(1974), pp.1-13.
- [7] Glover, F., "Tabu Search - Part I," *ORSA Journal on Computing*, Vol.1, No.3(1989), pp. 190-206.
- [8] Glover, F., "Tabu search - Part II," *ORSA Journal on Computing*, Vol.2, No.1(1990), pp. 4-32.
- [9] Glover, F. and M. Laguna, *Tabu Search*, Kluwer Academic Publishers, Dordrecht, 1997.
- [10] Hariri, A.M.A. and C.N. Potts, "Heuristics for Scheduling Unrelated Parallel Machines," *Computers & Operations Research*, Vol.18, No.3(1991) pp.323-331.
- [11] Horowitz, E. and S. Sahni, "Exact and Approximate Algorithms for Scheduling Non-identical Processors," *Journal of ACM*, Vol.23 (1976), pp.317-327.
- [12] Karp, R.M., *Reducibility among Combinatorial Problems - Complexity of Computer Computations*, Plenum Press, New York, 1972.
- [13] Laguna, M., J.W. Barnes and F. Glover, "Tabu Search Methods for Single Machine Scheduling Problem," *Journal of Intelligent Manufacturing*, Vol.2, No.1(1991), pp.63-74.
- [14] Lawler, E.L., J.K. Lenstra, A.H.G. Rinnooy Kan and D.B. Shmoys, *Sequencing and Scheduling : Algorithms and Complexity*, Handbooks in Operations and Management Science, Vol.4, 1993.
- [15] Lee, Y.H. and M. Pinedo, "Scheduling Jobs on Parallel Machines with Sequence-Dependent Setup Times," *European Journal of Operational Research*, Vol.100, No.3(1997), pp.464-474.
- [16] Monma, C.L. and C.N. Potts, "Analysis of Heuristic for Preemptive Parallel Machine Scheduling with Batch Setup Times," *Operations Research*, Vol.41, No.5(1993), pp.981-

- 993.
- [17] Ovacik, I.M. and R. Uzsoy, "Rolling Horizon Procedures for Dynamic Parallel Machine Scheduling with Sequence-Dependent Setup Times," *International Journal of Production Research*, Vol.33, No.11(1995), pp.3173-3192.
- [18] Ovacik, I.M. and R. Uzsoy, *Decomposition Methods for Complex Factory Scheduling Problems*, Kluwer Academic Publishers, 1997.
- [19] Pinedo, M., *Scheduling : Theory, Algorithms, and System*, Prentice Hall, New Jersey, 1995.
- [20] Schutten, J.M.J. and R.A.M. Leussink, "Parallel Machine Scheduling with Release Dates, Due Dates and Family Setup Times," *International Journal of Production Economics*, Vol.46/47(1996), pp.119-125.
- [21] Tang, C.S., "Scheduling Batches on Parallel Machines with Major and Minor Set-ups," *European Journal of Operational Research*, Vol.46, No.1(1990), pp.28-37.
- [22] Vasilescu, E.N. and A.D. Amar, "An Empirical Evaluation of the Entrapment Procedure for Scheduling Jobs on Identical Machines," *IIE Transactions*, Vol.15(1983), pp.261-263.
- [23] Vollmann, T.E., L.B. Berry, D.C. Whybark and F.R. Jacobs, *Manufacturing Planning and Control for Supply Chain Management*, McGraw-Hill, 2004.