經營科學
第24卷 第2號
2007年 11月

대안기계 스케쥴링 문제에 대한 라운딩 알고리듬*

황 학 진**

A rounding algorithm for alternate machine scheduling*

Hark-Chin Hwang**

■ Abstract ■

In this paper we consider an alternate m machine scheduling problem in which each job having at most two eligible machines should be assigned with the objective of makespan minimization. For this problem, we propose a $O(m2^m)$ time rounding algorithm with performance ratio at most 1.5. For a little general problem where each job can be processed in at most three machines, we prove that a polynomial time algorithm does not exist with performance ratio less than 1.5.

Keywords : Parallel machine scheduling, Approximation algorithm, Performance ratio, NP-Complete

1. 서 론

일반적으로 라인(line)생산에서는 한 종류의 제품을 전용설비를 통하여 생산한다. 그런데 고객의 기호가 다양해짐에 따라 하나의 라인에서 여러 종류의 주문을 수용할 수 있도록 변모되고 있다. 한 라인에서 다양한 종류의 제품을 생산하게 되면 일반적으로 제품단위당 생산성이 떨어지게 된다. 노무비, 재료비 등을 주로 고려하는 전통적인 원가계산 방식이 아니라 제품 하나의 생산과 관련된 셋업, 구매, 품질보증, 재고관리, 제품서비스 등과 관련된 활동원가를 고려하게 되면 회사전체의 수익률이

본 논문은 2006년도 한국경영과학회 추계학술대회(2006년 11월 17일) 우수논문상(응용부문) 수상논문으로 소정의 심사과정을 거쳐 게재 추천되었음.
논문접수일 : 2006년 12월 04일 논문게재확정일 : 2007년 06월 07일
* 이 논문은 2005년도 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임(KRF-2005-041-D00883).
** 조선대학교 산업공학과

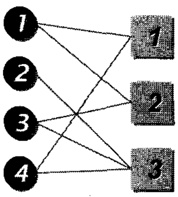
떨어지게 된다[18]. 따라서 고객의 다양한 기호를 만족시키기 위한 유연성과 회사 수익률 사이의 관계를 따져 라인당 생산될 제품의 개수를 가져갈 필요가 있다. 일반적으로 각 제품은 선호하는 하나의 전용라인을 갖게 되고 만약 그 라인의 주문이 폭주하면 대안 라인 또는 대안 기계(alternate line, alternate machine)으로 물량을 흘려 생산하는 경우가 많다. 본 논문에서는 각 제품당 작업 가능한 라인 또는 기계의 대수가 최대 2개 이하인 경우에 대한 작업할당에 대해서 고찰하고자 한다.

1.1 문제의 정의

n개의 작업과 m대의 기계가 주어진 대안기계 스케쥴링 문제를 보다 정확하게 정의하기 위하여 아래의 기호를 정의한다.

- p_j : 작업 j를 처리하는데 걸리는 시간, j = 1, 2, …, n.
- E_j : 작업 j를 처리할 수 있는 기계의 집합, 여기서 $E_j \subseteq \{1, 2, \cdots, m\}$이며, $|E_j| = 2$, j = 1, 2, …, n.
- M_i : 기계 i가 처리할 수 있는 작업들의 집합, 여기서 $M_i \subseteq \{1, 2, \cdots, n\}$이며 i = 1, 2, …, m.

작업과 기계의 작업가능 관계 및 이에 해당하는 E_j와 M_i가 <그림 1>에 나타나 있다.



$E_1 = \{1, 2\}$, $M_1 = \{1, 4\}$
$E_2 = \{3\}$, $M_2 = \{1, 3\}$
$E_3 = \{2, 3\}$, $M_3 = \{2, 3, 4\}$
$E_4 = \{1, 3\}$

〈그림 1〉 작업과 기계사이의 처리가능 형태

작업 *스케쥴*(schedule) 또는 일정계획이라 함은 n개의 작업을 m대의 기계에 작업 처리조건 (E_j)을 만족하며 나누어 할당한 것이다. 이 때, 제일 마지막으로 끝나는 작업의 완료시간을 그 스케쥴의 *메이크스팬*(makespan)이라 부른다. 본 논문의 목적함수는 메이크스팬을 최소화하는 것이다. 메이크스팬 최소화는 여러 대의 기계에 작업을 최대한 골고루 분산시켜, 기계활용율을 증대시키기 위한 방편으로 주로 이용된다.

만약, 처리가능 기계의 수에 대한 제약, 즉, $|E_j| = 2$이 없으면 대안기계 스케쥴링 문제는 일반적인 *다목적 병렬기계 스케쥴링 문제*(Multipurpose parallel machine scheduling problem)이다. 대안기계 스케쥴링 문제를 다목적 병렬기계 스케쥴링 문제 관점에서 정수계획법 모형을 정의하면 다음과 같다.

$$\min C_{\max}$$
subject to
$$\sum_{j \in M_i} p_j x_{ij} \le C_{\max} \quad i = 1, \cdots, m \tag{1}$$
$$\sum_{j \in E_j} x_{ij} = 1 \quad j = 1, \cdots, n \tag{2}$$
$$x_{ij} \in \{0, 1\} \quad i = 1, \cdots, m, \ j \in M_i \tag{3}$$

위 문제에서 변수 x_{ij}는 만약 작업 j가 기계 i에 할당되면 1을 그렇지 않으면 0의 값을 갖는다. 제약식 (1)은 각 기계에서 처리하는 토탈 시간이 최종작업처리시간 즉 메이크스팬을 넘을 수 없음을 나타낸다. 그리고 식 (2)와 식 (3)은 각 작업이 오직 한 기계에서만 처리되어야 함을 나타낸다.

일반적으로 이 문제는 1.5배 미만의 근사가 불가능한 것으로 알려져 있다[20]. 즉, P≠NP인 이상 최적 메이크스팬의 1.5배 미만의 스케쥴을 보장하는 폴리노미얼 (polynomial) 시간 알고리듬은 없다. 본 논문에서는 대안 기계스케쥴링 문제에 대하여 $O(m2^m)$ 시간 내에 1.5배 이하의 스케쥴을 보장하는 알고리듬을 제시한다. 또한, 각 작업의 처리 가능 기계의 대수가 3대 이하인 경우에도 1.5배 미만 근사가 불가능함을 보인다.

1.2 기존연구결과

본 연구와 관련된 기존의 연구를 체계적으로 고찰하기 위하여 일반적인 병렬기계스케쥴링 문제의

관점에서 조명하고자 한다. 병렬기계스케쥴링 문제는 반도체와 같은 제조산업[24]을 비롯하여 컴퓨터 시스템, 병원시스템[23] 등 다양한 분야에 응용 범위가 넓다. 먼저 문제 정의를 위한 표준기호 시스템을 보기로 한다. 일반적으로 병렬기계스케쥴링 문제는 세 기호 $\alpha|\beta|\gamma$로 나타내며, 여기서 α는 기계의 작업환경과 관련된 파라미터이고, β는 작업의 성격 그리고 γ는 목적함수를 나타낸다[7]. 여기서는 메이크스팬 감소가 주요한 요소이므로 $\gamma = C_{\max}$인 경우만 고려하기로 한다.

- 동일한 병렬기계 스케쥴링(identical parallel machine scheduling), $P||C_{\max}$: 모든 작업이 모든 기계에서 처리가능하다. 따라서 $E_j = \{1, 2, \cdots, m\}$이다. 이 문제는 동일한 성능의 기계가 여러 대 있는 경우이다.
- 다목적 병렬기계 스케쥴링(multipurpose parallel machine scheduling), $P|E_j|C_{\max}$: 생산라인 또는 기계가 유연해짐에 따라 각 작업 j를 처리할 수 있는 라인의 수, 즉, $|E_j|$의 크기가 증가한다. 대안기계스케쥴링 문제의 일반화된 문제이며 처리가능 기계 대수에 제한이 없다.
- 대안 병렬기계 스케쥴링(alternate parallel machine scheduling), $P|2E_j|C_{\max}$: 본 연구에서 다루는 문제이며, 다목적 병렬기계 스케쥴링 문제에서 $|E_j|$가 2이하인 특수한 경우이다.
- 무연관 병렬기계 스케쥴링(unrelated parallel machine scheduling), $R|p_{ij}|C_{\max}$: 작업과 기계사이의 처리시간이 어떠한 연관관계도 없이, 작업 j가 기계 i에서의 처리시간이 p_{ij}로 주어지는 경우이다. 우리의 문제 $P|E_j|C_{\max}$는 작업처리 시간을 다음과 같이 두면 $R|p_{ij}|C_{\max}$ 문제로 표현된다.

$$p_{ij} = \begin{cases} p_j & \text{if } i \in E_j, \\ \infty & \text{otherwise} \end{cases}$$

대안기계스케쥴링 문제에서 m = 2인 경우는 잘 알려진 NP-Complete 문제인 파티션(PARTITION) [5]과 같다. 즉, 대안기계 스케쥴링 문제에 대해서 폴리노미얼 시간 내에 최적해를 구하는 것(최소 메이크스팬을 갖는 스케쥴의 도출)은 거의 불가능하다. 따라서 비록 최적해는 아니지만 빠른 시간 내에 최적해에 근접하는 근사 알고리듬의 개발이 그 대안이 된다.

- $(1+\epsilon)$-근사 알고리듬

최적해가 C^*일 때, $(1+\varepsilon)C^*$시간 내에 모든 작업을 처리할 수 있는 스케쥴을 문제 사이즈의 폴리노미얼 시간 내에 도출하는 기법이 있다면 이 기법을 $(1+\varepsilon)$ - 근사 알고리듬이라 부른다. 여기서 $\varepsilon \geq 0$이다. 그리고 어떤 알고리듬이 $(1+\varepsilon)$ -근사이면 다른 말로 그 효능(performance ratio)이 $(1+\varepsilon)$이라고 부르기도 한다. 한편, 모든 $\varepsilon \geq 0$에 대하여 항상 $(1+\varepsilon)$ - 근사 알고리듬이 가능하다면 그러한 알고리듬은 특별히 PTAS(polynomial time approximation scheme)이라 부른다.

병렬기계 스케쥴링 문제들 중 가장 많이 연구된 문제가 동일한 병렬기계 스케쥴링(identical parallel machine scheduling) 문제이다. 1960년대에 처음으로 Graham에 의하여 연구되었다[6]. 이 문제에 대하여 단순할당(greedy) 알고리듬의 일종인 LS (List Scheduling)를 적용하여 항상 최적 메이크스팬의 $2 - 1/m$배 내의 스케쥴을 구할 수 있음을 증명하였고, 또한 LPT(Largest Processing Time first) 알고리듬이 $4/3 - 1/(3m)$ 근사 알고리듬임을 보였다[6]. 그 후에 메이크스팬에 대한 이진탐색(binary search) 방법을 이용한 MULTIFIT 알고리듬이 개발되었는데[3], 이 알고리듬의 효능은 13/11임이 밝혀졌다[4, 25]. 이 문제에 대한 PTAS는 [9]에서 최초로 개발되었고 좀더 효율적인 계산방법이 [10]에서 개발되었다.

한편, 기계들이 작업시작 시간 0에 가용하지 않을 경우에 대한 병렬기계 스케쥴링 문제에 대해서 [2]에서 다루었다. 이 문제는 많아야 m개 작업들의 처리가능기계의 수는 하나이고(즉, $|E_j| = 1$), 나머지 작업들은 모든 기계에서 처리가능(즉, $E_j = \{1, 2,$

$\cdots, m\}$)한 다목적 병렬기계 스케쥴링 문제의 좀 특수한 경우이다. 이 문제에 대하여 MULTIFIT이 적용되어 그 효능이 9/7임이 밝혀졌고[2], PTAS가 개발되었다[12]. 또한 기계들이 시작 시간 0뿐만 아니라 일반적으로 특정 시간의 구간동안 사용 불가능한 경우의 동일한 병렬기계 스케쥴링 문제에 대해서 LPT를 적용한 결과, 그 효능이 적어도 전체 기계의 반은 살아 있을 경우 2임이 증명되었다[11].

다목적 병렬기계 스케쥴링(multipurpose parallel machine scheduling) 문제는 Azar에 의하여 연구되기 시작했다[1]. 그들이 제시한 알고리듬은 LS와 유사한 것이며 항상 최적 메이크스팬의 $\log_2 2m$배 내의 스케쥴을 보장함을 증명하였다. 한편, 메이크스팬을 결정하는 기계에 할당된 작업의 특성에 기반하여 LS 알고리듬의 성능을 보다 면밀히 [16]에서 분석하였다. 그 결과 최적해의 $\log_2 \frac{4}{\lambda}m - \frac{1}{\lambda}$배 내의 스케쥴을 항상 도출함을 보였다. 여기서 λ는 최종적으로 끝나는 작업이 갖는 처리가능한 기계의 수이다. 또한 LPT를 적용할 경우에는 조금 더 향상되는데 $\log_2 \frac{4}{1+\lambda}m$의 효능을 보임이 증명되었다[14].

이 문제는 선형계획을 이용하여 해결가능한데 [20]에서 2-근사 알고리듬을 개발하였다. 또한 이 문제에 대하여 1.5미만의 근사 알고리듬이 존재하지 않음을 이론적으로 증명하였다. 따라서 지금까지 기존의 2-근사 알고리듬을 뛰어넘어 1.5와 2사이의 (예를 들어 1.75) 근사알고리듬이 존재할 것인가에 대한 의문이 근 20년간 지속되어 왔다. 최근에 선형계획에 기초하여 보다 정교한 라운딩 알고리듬이 개발되었는데 그 효능이 $2-1/m$이다[22]. 기계의 대수가 많을 때, 일반적으로 이 알고리듬도 2-근사 알고리듬임을 알 수 있다. 이 문제의 좀 특이한 형태로 각 작업 j의 처리가능 기계집합 E_j가 $\{1, 2, \cdots, k\}$ 기계 1번부터 임의의 $k \leq m$까지 주어지는 경우이다(GoS, Grade of Service eligibility). 이 모델은 고객주문의 중요도에 따라 기계를 배정할 때 활용되며, LPT를 적용할 때 $2-1/(m-1)$의 근사가 가능하다는 것이 밝혀졌다[15]. 위 문제보다 조금 일반적인 형태인 E_j가 기계 인덱스의 구간으로 주어지는 경우에 대해서 연구가 진행되었다[13](즉, $E_j = \{i_1, i_2, \cdots, i_k\}$일 때, $i_2 = i_1 + 1, \cdots, i_k = i_{k-1} + 1$). MULTIFIT와 유사한 알고리듬 MFFP가 개발되었는데 이는 2-근사 알고리듬으로 빠른 시간 내에 해를 도출하는 것으로 알려져 있다.

무연관 병렬기계 스케쥴링 문제에 대하여 효능 m의 알고리즘이 개발된 후[17], 2 및 $2-1/m$ 효능의 알고리듬이 각각 개발되었다[20, 22].

2. 선형계획 완화

원래 문제 식 (1)~식 (3)에서 제약식 (3)을 완화(relax)하여 x_{ij}의 값이 0, 1 이진변수가 아닌 0이상의 실수 값을 갖도록 완화하자. 즉 식 (3)을 아래의 제약식으로 대체하면

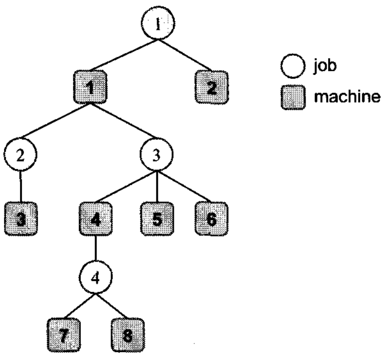
$$x_{ij} \geq 0, \ i = 1, \cdots, m \ j \in M_i$$

선형계획 완화(linear programming relaxation) 문제가 된다.

변수 $C_{\max}$를 포함한, 총 변수의 개수를 v라고 둘 때, 이 선형계획 문제가 갖는 가능해(feasible region)의 특성을 살펴보면 다음과 같다[20, 21].

- 제약식의 총 개수는 $v+m+n-1$이다.
- 변수 $C_{\max}$에 대한 제약은 없으므로, 가능해의 모서리점(extreme point)은 $v-1$개의(등가, equality) 제약식으로 정의된다.
- 따라서, 임의의 모서리점 x에 대하여 $m+n$개의 변수만 제외하고 모두 0의 값을 갖는다.
- 최적해에서 $C_{\max} > 0$임으로, 변수 x_{ij} 중 0이 아닌 개수는 $m+n-1$개이다.
- 작업 j와 관련된 변수 x_{ij}, $i \in E_j$에 대해서 적어도 두 변수(예를 들어 x_{1j}, x_{2j})이상이 0보다 큰 값을 가질 때, 작업 j는 비정수형 값을 갖는다. 즉 두 대 이상의 기계에 나누어져 할당된다.
- 그러므로 비정수형 값을 갖는 작업의 개수는 최대 $m-1$개이다.

작업과 기계 사이의 할당관계를 그래프를 이용하여 표시할 수 있다. 선형계획 완화 문제의 해 x에서 비정수형 값을 갖는 x_{ij} 값들에 대하여 기계노드 i와 작업노드 j의 연결로 이루어지는 그래프를 보면 트리(tree) 형태의 그래프가 만들어 진다[20, 22]. 트리 형태를 갖는 이유는 비정수형 값을 갖는 작업의 개수는 최대 m-1개라는 사실로부터 쉽게 알 수 있다. <그림 2>에서 8대의 기계에 대해 작업 4개가 비정수형 값을 가질 때, 작업-기계 그래프의 예가 나와 있다.



〈그림 2〉 비정수형 작업과 기계에 대한 그래프

그러면 기존의 라운딩 알고리즘을 이용하여 다음의 결과를 쉽게 얻을 수 있다.

정리 1 : 선형계획 완화문제의 해로부터 비정수형 값에 해당하는 작업들 중 가장 큰 작업시간을 p_{max}라 두자. 그러면 [20] 또는 [22]의 라운딩 알고리즘을 이용하여 $C^* + p_{max}$이내의 메이크스팬을 갖는 스케쥴을 만들 수 있다.

모든 작업의 처리시간이 최적메이크스팬 보다 클 수 없다. 즉, $p_{max} \leq C^*$이다. 이에 기초하여 [20]에서는 항상 $2C^*$를 보장하는 스케쥴을 만들 수 있었고, [20]의 알고리듬을 좀더 정교하게 만들어 [22]에서는 $(2-1/m)C^*$을 보장하는 스케쥴을 만들 수 있었다. Lenstra et al.[20]은 정수계획법 문제식 (1)~식 (3)과 조금 다른 문제를 풀었다. C_{max}값을 변수로 두지 않고, 임의의 상수 값으로 두고, 식 (1)~식 (3)을 만족하는 가능해(feasible solution)가 있는 가를 테스트하였다. 만약, 가능해가 존재하면, C_{max}값을 줄이고, 그렇지 않으면 늘리는 이진탐색의 방법을 통하여 최적 메이크스팬을 구하였다. 이 방법에서는 C_{max}가 변수가 아니므로, 변수 x_{ij}들 중 0이 아닌 개수는 최대 $m+n$개임을 알 수 있다. 따라서 이에 대한 작업-기계 그래프는 트리가 아니라, 트리에서 싸이클이 많아야 한 개 있는 그래프로 표현되며, 이러한 특성으로부터 효능 2의 알고리듬이 개발되었다. 한편, [22]에서는 C_{max}를 변수로 취급하고, 최적해에서 이 값이 0보다 크다는 점을 이용하여[21], 0이 아닌 변수 개수는 최대 $m+n-1$개라는 사실을 확인하였다. 따라서 작업-기계 그래프는 항상 트리 형태이며, 이와 같은 구조를 바탕으로 효능이 $2-1/m$인 알고리듬을 제시하였다.

정리 1로부터 우리는 p_{max} 값만 줄어든다면 최적의 2배 미만의 스케쥴을 만들 수 있음을 알 수 있다. 즉, 만약 p_{max} 값이 $0.5C^*$ 보다 크지 않다면 우리는 최적해의 1.5배 이하의 스케쥴을 만들 수 있다. p_{max} 값이 $0.5C^*$ 보다 작으려면 비정수형 값을 갖는 작업의 크기가 모두 $0.5C^*$ 이하이어야 한다. 시간 $0.5C^*$를 기준으로 작업을 두 유형으로 구분하고자 한다. 작업의 크기가 최적 메이크스팬의 1/2배 보다 큰 작업을 유형-B(Big)라 부르기로 하고, 그렇지 않은 작업들을 유형-S(Small)라 부르기로 한다. 이러한 작업의 분할(partition)에 기초하여 1.5-근사 알고리듬을 개발하고자 한다.

3. $O(m2^m)$ 시간 1.5-근사 알고리듬

먼저, 유형-B 작업들에 대한 성질을 살펴보자.

3.1 유형-B 작업들에 대한 스케쥴의 성질

최적해에서 유형-B가 갖는 성질들 중 가장 일

반적인 것은 유형-B의 두 작업이 한 기계에 함께 처리될 수 없다는 것이다. 왜냐하면 유형-B 두 작업의 처리시간의 합은 최적 메이크스팬 보다 크기 때문이다. 이 성질로부터 유형-B 작업의 개수는 최대 m개임을 또한 알 수 있다. 다음으로 대안기계 스케줄링 문제에서의 유형-B 작업들의 성질을 살펴보자.

각 작업을 처리할 수 있는 기계의 대수가 최대 2대임으로 유형-B 작업들과 기계 사이의 연결 관계를 그래프로 나타내면 각 *연결 컴포넌트*(connected component)는 트리 또는 싸이클이 많아야 하나밖에 없는 트리 형태가 된다. 여기서 작업과 기계사이의 그래프에서 모든 두 노드사이에 경로(path)가 있으면 이 그래프는 '연결(connect)되었다'라고 부르고, 어떤 그래프가 여러 개의 연결된 부분으로 나누어져 있을 때, 각 연결된 부분을 컴포넌트(component)라 부른다. 유형-B 작업-기계 그래프의 한 연결 컴포넌트에서 유형-B 작업의 개수가 k이고, 기계의 개수가 l이라 두자. 이때, 유형-B 작업들은 서로 같은 기계에 할당될 수 없으므로 k는 l보다 클 수 없다. 즉, 그래프가 트리이거나 싸이클이 하나 이하인 트리이다. <그림 3>(a)와 <그림 3>(b)는 유형-B 작업과 기계사이에 나올 수 있는 가능한 그래프 형태이지만, <그림 3>(c)는 불가능한 형태이다.

따라서 우리는 다음의 사실과 함께

$$k = l \text{ 또는 } k = l-1$$

다음의 정리를 쉽게 증명할 수 있다.

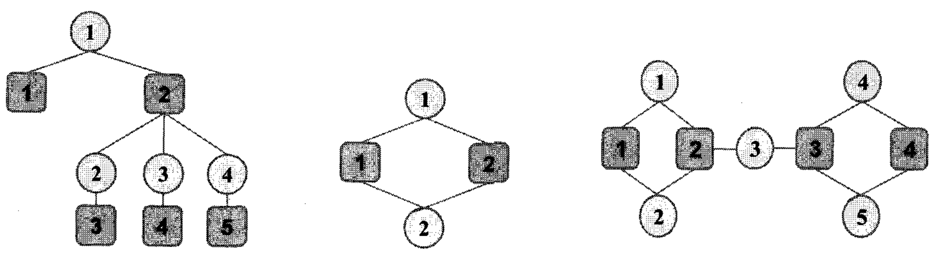
정리 2 : 유형-B 작업과 기계 그래프의 한 연결컴포넌트에서 유형-B 작업이 k개일 때, 기계에 할당되는 경우의 수는 k개 이하이다.

3.2 1.5-근사 알고리듬

1.5-근사 알고리듬을 가능하게 하려면 유형-B 작업의 배치를 선행하고, 나머지 작업들에 대해서는 선형계획 완화 문제를 풀어 해결할 수 있다. 유형-B 작업을 미리 배치하게 되면, 그 사실을 선형계획 모형에 반영하여야 한다. 유형-B 작업이 배치된 후 각 기계 i에 할당된 물량(처리시간)을 s_i라 두자. 그리고 유형-S의 작업개수를 n이라 두자. 이때의 선형계획 완화문제는 다음과 같다.

$$\begin{aligned} &\min C_{\max} \\ &\text{subject to} \\ &\quad \sum_{j \in M_i} (s_i + p_j x_{ij}) \le C_{\max} \qquad i = 1, \cdots, m \\ &\quad \sum_{j \in E_j} x_{ij} = 1 \qquad\qquad\qquad j = 1, \cdots, n \\ &\quad x_{ij} \ge 0 \qquad\qquad\qquad\quad\;\; i = 1, \cdots, m,\ j \in M_i \end{aligned}$$

처음부터 어느 작업이 유형-B 작업인지 알 수 없다. 이러한 정보의 부재를 해결하기 위하여 모든 가능성을 고려하여야 한다(enumerative computation). 즉, 유형-B 작업의 개수가 0, 1, ⋯, m인 모



(a) (b) (c)

〈그림 3〉 유형-B 작업과 기계 사이의 그래프

든 경우에 대하여 문제를 풀어야 한다.

알고리듬

단계 0 : (초기화). $k = 0$. $C* = \infty$.

//처음에는 유형-B 작업의 개수를 0이라고 둠.

단계 1 : 작업 1, 2, …, k를 유형-B 작업이라 두고, 이들 작업과 기계 사이의 관계를 그래프로 표시한다.

단계 2 : 유형-B의 작업이(각 연결 컴포넌트 별로) 할당되는 각 경우에 대해서

2.1 : 위 선형계획완화문제를 풀고,

2.2 : [22]의 라운딩 알고리듬을 적용한다.

2.3 : 그 결과 나온 스케쥴의 메이크스팬이 C일 때, $C < C*$이면, $C* = C$로 놓는다.

단계 3 : (종료). $k > m$이면 종료하고, 그렇지 않으면 $k = k + 1$로 두고, 단계 2로 간다.

유형-B 작업과 기계사이의 그래프에서 연결 컴포넌트의 개수가 k개이고, 각 연결 컴포넌트에서 작업의 개수가 $u(i)$, $i = 1, 2, \cdots, k$라고 가정하자. 그러면, 유형-B 작업이 배치될 수 있는 경우의 수는 정리 2에 의하여 총 $u(1) \times u(2) \times \cdots \times u(k)$개이다. 그런데 유형-B 작업의 개수는 최대 m개 이므로, 위 알고리듬의 단계 2에서 유형-B작업의 배치가능 개수는 $u(1) \times u(2) \times \cdots \times u(m)$개 이다. 각 연결 컴포넌트에서 유형-B 작업의 할당 경우의 수는 2^m이다. 따라서 우리는 유형-B 작업의 배치가능 개수는 많아야 $m2^m$임을 쉽게 알 수 있다. 그리고 나머지 유형-S 작업들에 대한 선형계획 완화문제의 비정수형 해에 대해서 기존의 라운딩 알고리듬을 적용하면 정리 1에 의하여 $C* + p_{\max}$의 메이크스팬을 갖는 스케쥴을 만들 수 있다. 여기서 $p_{\max}$는 유형-S 작업들 중 최대 작업시간을 가리키며, 정의에 따라 이 값은 $0.5C*$이하이다. 따라서 우리는 다음과 같은 결과를 알 수 있다.

정리 3 : m대 기계의 대안기계 스케쥴링 문제에 대하여 $O(m2^m)$ 시간내에 최적 메이크스팬의 1.5배 이하의 스케쥴을 도출하는 알고리듬이 존재한다.

만약 유형-B작업의 개수가 m개 일 때에는 이들이 배치될 수 있는 경우의 수가 m개 이하임으로 다음의 결과도 쉽게 증명할 수 있다.

정리 4 : m대 기계의 대안기계 스케쥴링 문제에서 최적 메이크스팬의 0.5배 보다 큰 작업의 개수가 m개 일 때, 최적 메이크스팬의 1.5배 이하의 스케쥴을 도출하는 폴리노미얼 알고리듬이 존재한다.

4. 근사의 한계

각 작업을 처리할 수 있는 기계가 2대 이하인 문제에 대하여 비록 폴리노미얼 알고리즘은 아닐지라도 1.5배 이하의 스케쥴을 도출하는 방법에 대해서 살펴 보았다. 그렇다면 이와 유사한 방법의 근사 또는 폴리노미얼 시간내에 최적해의 1.5배 이하의 스케쥴을 보장하는 알고리즘이 존재할까? P = NP가 아닌 이상, 작업 처리기계 대수가 3이하로 제한될지라도, 이 문제($P|3E_j|C_{\max}$)에 대한 1.5미만 효능의 알고리듬은 존재하지 않음을 보이고자 한다. 증명은 Lenstra et al.[20]의 정리를 이용하고자 하며, 기존의 잘 알려진 NP-Complete 문제인 3차원 매칭-한계 3으로부터 환원하고자 한다. 참고로 Lenstra et al.[20]에서는 가장 어려운 문제인 무연관 병렬기계 스케쥴링 문제에서 1.5배 미만의 근사 알고리듬이 없음을 증명하였다.

3차원 매칭-한계3(MB-3DM, Maximum Bounded 3-Dimensional matching)(Garey and Johnson, [5])

- 인스턴스(instance) : 서로 교차하는 원소가 없는 세 개의 집합 $X = \{x_1, \cdots, x_n\}$, $Y = \{y_1, \cdots, y_n\}$, $Z = \{z_1, \cdots, z_n\}$에 대하여 하나의 트리플 $T_i = \{x_j, y_k, z_l\}$는 각 집합으로부터 하나씩 선택된 원소로 구성된다. F를 m개의 트리플로 구성된 집합이라고 두자 $(m > n)$. 여기서 집합 X, Y, Z

의 각 원소는 F의 모든 트리플에서 3번 이하 포함된다.

- 질문(question) : 이때, 집합 F에 매칭이 존재하는가? 즉 다음을 만족하는 F'이 존재하는가? (여기서 $|F'| = n$이다).

$$\cup_{T_i \in F'} T_i = X \cup Y \cup Z$$

위에서 정의된 3차원매칭 문제에 대하여 기계의 대수가 m이고 작업의 개수가 $2n+m$인 대안기계 스케쥴링 문제를 먼저 정의하기로 한다. 여기서 기계 i는 트리플 T_i, $i = 1, \cdots, m$에 해당한다 X의 한 원소 x_j를 포함하는 모든 트리플 $T_i = \{x_j, -, -\}$을 유형 - x_j라 부르기로 한다. 그리고 t_j를 F에서 유형 - x_j인 트리플의 개수라 두자. MB-3DM의 정의에 따라 t_j는 3이하이다. 먼저 각 유형-x_j에 대해서 총 t_j-1개의 작업 $a_k(j)$, $k=1, \cdots, t_j-1$을 만든다. 그러면, 모든 작업 $a_k(j)$의 총 개수는 $m-n$개임을 쉽게 알 수 있다. 두 집합 Y, Z의 각 원소에 대하여 $b(1), \cdots, b(n)$, 그리고 $c(1), \cdots, c(n)$의 $2n$개의 작업을 정의한다.

다음으로 각 작업의 처리시간과 처리가능 기계에 대해 살펴보자. 먼저, 작업 $a_k(j)$은 처리시간 2를 갖고 작업 $b(j)$와 $c(k)$는 처리시간 1을 각각 갖는다. 또한 각 작업의 처리가능 기계 i는 다음과 같다.

$i \in E(a_k(j))$, 만약 트리플 $T_i = (x_j, -, -)$인 경우,
$j = 1, \cdots, n, \ k = 1, \cdots, t_j - 1$

$i \in E(b(k))$, 만약 트리플 $T_i = (-, y_k, -)$인 경우,
$k = 1, \cdots, n$

$i \in E(c(l))$, 만약 트리플 $T_i = (-, -, z_l)$인 경우,
$l = 1, \cdots, n$

위에서 정의된 각 집합 E_j를 살펴보면, 작업 $a_k(j)$를 처리할 수 있는 기계는 오직 한 대임을 알 수 있다. MB-3DM의 정의에서 Y, Z의 각 원소는 F의 모든 트리플에서 3번 이하 포함된다. 즉, 작업 $b(k)$, $c(l)$을 처리할 수 있는 기계의 대수는 많아야 3대이다.

정리 5 : 처리가능 기계의 대수가 3대 이하인 다목적 병렬기계문제($P|3E_j|C_{\max}$)에 대하여 P$\neq$NP인 경우 1.5미만 효능의 폴리노미얼 시간 알고리듬은 존재하지 않는다.

(증명) 일반적인 MB-3DM 문제가 위에서 정의한 대안기계 스케쥴링 문제(각 작업의 처리시간이 1 또는 2)로 환원됨을 먼저 보이고자 한다.
(->) 먼저 매칭이 존재하면, 메이크스팬이 2인 스케쥴이 있음을 보인다. 각 트리플 $T_i = \{x_j, y_k, z_l\}$에 대하여 작업 $b(k)$, $c(l)$을 할당한다. 나머지 t_j-1개의 유형-x_j에 해당하는 작업 $a_k(j)$은 t_j-1개의 유형-x_j에 해당하는 기계에 각각 할당한다. 그러면 메이크스팬이 2인 스케쥴이 만들어 진다.
(<-) 다음으로 메이크스팬이 2인 스케쥴이 있으면 매칭이 존재함을 증명하고자 한다. 작업시간이 2인 작업 $a_k(j)$는 유형-x_j에 해당하는 기계들 중 하나에 할당되어 있어야 한다. 따라서 유형-x_j의 기계들 중 $a_k(j)$와 같은 작업을 처리하지 않는 기계는 정확하게 한 개 존재한다. 그 기계를 i라 두자. 그러면 기계 i에 해당하는 트리플 $T_i = \{x_j, y_k, z_l\}$에 해당하는 두 작업 $b(k)$, $c(l)$이 할당되어 있어야 한다. 각 작업 $b(k)$, $c(l)$, k, l = 1, $\cdots$, n은 정확하게 한 번씩 할당되므로, 이들 작업이 할당된 기계에 해당하는 트리플을 모으면 우리가 원하는 매칭을 얻을 수 있다.

그런데 MB-3DM 문제가 NP-Complete임으로 최적 매칭을 구하는 것은 불가능하다. 다시 말해서 대안기계 스테쥴링 문제에서 메이크스팬이 2인 스케쥴을 구하는 것은 불가능하다. 그런데 메이크스팬이 3인 스케쥴은 쉽게 구할 수 있다. 따라서 최적메이크 스팬의 1.5배 미만의 스케쥴을 보장하는 알고리듬은 일반적으로 존재하지 않는다.

5. 결 론

본 연구에서는 대안기계 스케쥴링 문제에 대해서 선형계획 완환에 기초한 라운딩 알고리듬을 제

안하였다. 이 알고리듬은 $O(m2^m)$ 계산 시간 내에 최적 메이크스팬의 1.5배 이하의 스케쥴을 보장함을 증명하였다. 비록 작업 처리가능 기계의 대수가 3으로 제한될지라도 여전히 어려운 문제임이 판명되었다. 이 문제에 대해서도 1.5미만 효능의 폴리노미얼 알고리듬이 없음을 증명하였다.

한편, 본 논문의 결과가 일반적인 다목적 병렬기계 스케쥴링 문제에도 적용될 것인가가 향후 연구 과제이다. 또한 적어도 대안기계 스케쥴링 문제에 대해서 1.5효능의 폴리노미얼 알고리듬이 존재할 것인가에 대해서도 풀어야 할 과제이다.

참 고 문 헌

[1] Azar, Y., J. Naor, and R. Rom, The competitiveness of On-Line Assignments, *J. Algorithms*, Vol.18(1995), pp.221-237.

[2] Chang S.Y. and H.-C. Hwang, The worst-case analysis of the MULTIFIT algorithm for scheduling nonsimultaneous parallel machines, *Discrete Applied Mathematics,* Vol. 92(1999), pp.135-147.

[3] Coffman Jr E.G., M.R. Garey, and D.S. Johnson, An application of bin-packing to multiprocessor scheduling, *SIAM J. Comput.*, Vol.7(1978), pp.1-17.

[4] Friesen D.K., Tighter bounds for the multi processor scheduling algorithm, *SIAM J. of Comput.*, Vol.13(1984), pp.35-59.

[5] Garey M.R. and D.S. Johnson, *Computers and Intractability : A Guide to the theory of NP-Completeness*, Freeman, San Francisco, 1979.

[6] Graham R.L., Bounds on multiprocessor timing anomalies, *SIAM J. Appl. Math*, Vol. 17(1969), pp.263-269.

[7] Graham, R.L., E.L. Lawler, J.K. Lenstra, and A.H.G. Rinnooy Kan, Optimization and Approximation in Deterministic Machine Scheduling : A Survey, *Annals of Discrete Mathematics*, Vol.5(1979), pp.287-326.

[8] He Y., H. Kellerer, and V. Kotov, Linear Compound Algorithms for the Partitioning Problem, *Naval Research Logistics,* Vol.47 (2000), pp.593-601.

[9] Hochbaum D.S. and D. Shmoys, Using dual approximation algorithms for scheduling problems : Theoretical and practical results, *J. ACM*, Vol.34(1987), pp.144-162.

[10] Hochbaum D.S., *Approximation Algorithms for NP-Hard Problems,* PWS PUBLISHING COMPANY, Boston, (1997), 370-371.

[11] Hwang H.-C. and S.Y. Chang, Parallel machines scheduling with machine shutdowns, *Computers and Mathematics with Applications*, Vol.36(1998), pp.21-31.

[12] Hwang H.-C, A PTAS for nonsimultaneous parallel machine scheduling, Journal of the Korean Society of Maintenance Management, Vol.3(2003), pp.181-193.

[13] Hwang H.-C. and G. Kim, 2-Approximation Algorithm for Parallel Machine Scheduling with Consecutive Eligibility, *KIIE,* Vol.29(2003), pp.190-196.

[14] Hwang H.-C, LPT Scheduling for Multipurpose Machines, *IE Interfaces*, Vol.16(2003), pp.132-137.

[15] Hwang H.-C, K. Lee, and S.Y. Chang, Parallel machine scheduling under a grade of service provision, *Computers & Operations Research*, Vol.31(2004), pp.2055-2061.

[16] Hwang H.-C, S.Y. Chang, and Y. Hong, The Posterior Competitiveness for List Scheduling Algorithm on Machines with Eligibility Constraints, *Asia-Pacific Journal of Operational Research*, Vol.1(2004), pp.1-9.

[17] Ibarra O.H. and C.E. Kim, Fast approximation algorithms for the knapsack and sum of subset problems, *J. ACM*, Vol.22 (1975), pp.463-468.

[18] Kaplan R.S. and R. Cooper, *Cost & Effect*, Havard Business School Press, Boston, Massachusetts, 1997.

[19] Kellerer H., U. Pferschy, A New Fully Polynomial Approximation Scheme for the Knapsack Problem, *Journal of Combinatorial Optimization,* Vol.3(1999), pp.59-71.

[20] Lenstra J.K., D.B. Shmoys, and E. Tardos, Approximation Algorithms for Scheduling Unrelated Parallel Machines, *Mathematical Programming,* Vol.46(1990), pp.259-271.

[21] Potts C.N., Analysis of a linear programming euristic for scheduling unrelated parallel machines, *Discrete Appl. Math.* Vol. 10(1985), pp.155-164.

[22] Shchepinal E.V. and N. Vakhania, An optimal rounding gives a better approximation for scheduling unrelated machines, *Operations Research Letters*, Vol.33(2005), pp. 127-133.

[23] Vairaktarakis, G.L. and X. Cai, The Value of Processing Flexibility in Multipurpose Machines, *IIE Transactions*, Vol.35(2003), pp.763-774.

[24] Yu, L., Scheduling of unrelated parallel machines : an application to PWB manufacturing, *IIE Transactions*, Vol.34(2003), pp. 921-931.

[25] M. Yue, On the exact upper bound for the MULTIFIT processor scheduling algorithm, in *Operations Research in China,* M. Yue (ed.), of Annals of Operations Research, Baltzer, Basel, Switzerland, Vol.24(1990), pp. 233-259.