韓國經營科學會誌
第23卷 第1號
1998年 3月

최대 지연시간을 고려한 ET 모델에서의 단일 기계 일정계획*

조성아** · 조충호** · 이동훈** · 김채복***

Single Machine Scheduling with Maximum Allowable Tardiness in ET Model

Sunga Cho** · Choongho Cho** · Dong Hoon Lee** · Chae-Bogk Kim***

Abstract

This paper addresses the problem of scheduling a set of jobs with a common due date on a single machine. The objective is to minimize the sum of the earliness and tardiness of jobs subject to $T_{max} \leq \Delta$ for $\Delta \geq 0$. Properties for the MAD/T_{max} problem are found and the problem is shown to be NP-complete in the ordinary sense. According to the range of Δ, the problem can be solved in polynomial time. Also, some special cases where an optimal schedule is found in polynomial time are discussed.

1. 서 론

산업체에서 발생하는 많은 문제중 하나는 작업의 순서를 결정하는 것이며 특히 최적의 스케줄을 발견하는 문제는 집중적으로 연구되어지고 있는 분야 중 하나이다. 수 많은 스케줄링 모델 중에서 작업의 완료시간 (completion time)이 요구된 납기 (due date)와 다를 경우 페널티 (penalty)를 부과하는 earliness/tardiness (ET) 모델에 대한 연구는 최근에 많은 학자들이 관심을 가지는 분야이다.

지난 수십년간 스케줄링 연구는 주로 작업의 완료시간이 늦어지면 척도 (measure)의 값

* 이 논문은 1996년도 학술진흥재단의 공모과제 연구비에 의하여 연구되었음.
** 고려대학교 전산학과
*** 한국교원대학교 기술교육학과

이 증가하는 평균 완료시간과 평균 지연시간 등과 같은 regular measure에 초점을 맞추었다. 특히, 평균 지연시간은 일찍 끝난 작업들에 페널티를 부과하지 않음에도 불구하고 작업들이 납기를 얼마나 잘 지키느냐하는 고객에 대한 신뢰성을 나타내는 척도로 여겨져 왔으며 이를 최소화하는 스케쥴을 발견하기 위하여 동적 계획법 (dynamic programming)이나 분지한계법 (branch and bound)이 연구되었다.

그러나, 1980년대에 일본에서 유래된 "just-in-time"의 개념이 전세계에 소개되면서 납기보다 늦게 끝난 작업들뿐만 아니라 일찍 끝난 작업들에도 페널티를 부과하는 ET 모델의 스케줄링 문제들에 학자들이 관심을 갖기 시작했다. 도요타 자동차공장에서 사용되어진 "just-in-time"의 개념은 필요한 양 (quantity)과 질 (quality)의 제품이 적재적소에 생산되어지거나 공급되어져야 한다는 것이다.

Earliness와 tardiness를 동시에 고려하는 스케줄링에서의 척도는 작업의 완성시간이 늦어져도 반드시 증가하지는 않기 때문에 nonregular measure이다. 그 이유는 earliness가 nonregular measure이므로 earliness와 tardiness의 합으로 구성되는 척도도 nonregular measure이다. 납기보다 먼저 작업을 마치는 earliness의 중요성을 강조하는 예들은 상하기 쉬운 유제품, PERT/CPM관리, file organization 등이 있으며 이는 재고 관리의 어려움과 재고 비용과 관계가 있다 [1, 2, 4, 6]. 일반적으로 tardiness가 고객과의 신뢰도, 납기 준수 여부와 관계가 있다면 earliness는 재고와 관계가 있으며 ET 모델에서의 스케줄링에 관한 연구는 Baker와 Scudder에 [3] 의하여 목적함수, 제약조건, 주어진 가정, 기계의 수에 따라 잘 정리되어 있다.

납기를 지키지 못한 작업 중 가장 정도가 큰 작업의 지연시간을 표시하는 최대지연시간 (maximum tardiness : T_{max})은 실제 상황에서 매우 중요한 척도이다. 신뢰성 문제를 표시할 수 있는 이유로 스케줄링의 여러 가지 다목적 의사 결정 문제들에 자주 사용되어지는 척도이며 회사가 고객에 대한 신용의 정도를 나타내는 기준으로도 사용되어진다.

스케줄링 문제에서 다목적으로 의사결정을 하는 연구는 많이 수행되어져 있으며 earliness와 tardiness를 동시에 고려하면서 다른 척도와의 관계를 밝힌 연구는 납기결정과 관련하여 많이 수행되고 있다 [3]. 그러나 의사결정자에게 일정 관리에 있어서 중요한 척도중의 하나인 최대지연시간을 ET모델에서 함께 고려한 연구는 문제의 복잡성 (complexity) 때문에 수행되어져 있지 않다. ET모델에서도 목적함수의 형태에 따라 최적해를 찾는 것이 다항시간내에 해결되지 않는 경우가 있다. 그러나, 단일기계에서의 평균 절대편차 (mean absolute deviation: MAD)를 최소화하는 ET모델은 다항시간내에 해결할 수 있는 문제이다.

본 논문에서는 MAD/T_{max} 문제, 즉 주어진 최대 허용지연시간(maximum allowable tardiness) 이내에 작업들이 완료되는 스케줄 중에서 평균 절대편차를 최소화하는 스케줄을 발견하는 문제를 다룬다. 2절에서는 MAD/T_{max} 문제와 이 문제의 목적함수를 정의하고 이에 따르는 몇가지 성질들을 다룬다. 3절에서는 MAD/T_{max} 문제중 다항시간이내에 문제를 해결할 수 있는 경우에 대해 생각해보고 4절에서는 MAD/T_{max} 문제가 NP-complete인 경우와 이에 대한 증명을 보인 후 자기발견적 (heuristic) 알고리즘을 제안하겠다.

2. 최대지연시간을 고려한 MAD 문제(MAD/T_{max} 문제)

스케줄될 작업들이 1, 2, … n, 작업 j의 가공시간이 p_j 그리고 모든 작업에대한 납기가 공통 납기 d로 표시되며, 작업들은 분리되어서 가공할 수 없다고 가정한다. 또한, 작업들은 가공시간에 따라 $p_1 \le p_2 \cdots \le p_n$ 로 분류되어져 있으며, 작업은 0인 시점에서 가공준비가 되어있다고 생각하자. 임의의 스케줄에서 작업 j의 완료시간이 C_j이며 j번째 작업의 완료시간을 $C_{[j]}$라고 할 때, 작업 j의 earliness와 tardiness는 각각 E_j와 T_j로 표시되며 다음과 같이 정의된다:

$$E_j = \max(0, d - C_j),\ T_j = \max(0, C_j - d).$$

본 논문에서 최소화하려는 목적함수는 위의 정의를 이용하여 다음과 같이 표현할 수 있으며 이 때의 임의의 스케줄 S의 목적함수 값을 $Z(S)$라 한다.

$$Z(S) = \sum_{j=1}^{n} abs(C_j - d) = \sum_{j=1}^{n} (E_j + T_j).$$

MAD/T_{max} 문제, 즉 주어진 최대 허용지연시간이내에 평균 절대편차를 최소화하는 단일기계에서의 최적 스케줄을 발견하는 문제를 해결할 때, 최대 허용지연시간이 최적 스케줄에 미치는 영향을 분석하기 위하여 납기 d는 충분히 크다고 가정한다. Δ가 주어진 최대 허용지연시간을 의미할 때 이 문제는 다음과 같이 정의될 수 있다.

$$\text{Minimize} \quad \sum_{j=1}^{n} abs(C_j - d)$$
$$\text{subject to } T_{\max} \le \Delta.$$

본 논문에서 사용되는 기호들은 다음과 같다:

$$MS \equiv \sum_{j=1}^{n} p_i$$
$$\eta \equiv p_{n-1} + p_{n-3} + \cdots$$
$$\theta \equiv p_{n-1} + p_{n-3} + \cdots$$

$T^*_{\max}$ ≡ 최적 스케줄에서의 최대허용 지연시간

만일 $d + \Delta <$ MS이면 주어진 제약조건을 만족하는 해가 없으므로 $d + \Delta \ge$ MS이라 가정한다. 임의의 스케줄에서 납기보다 빠르거나 같이 완료되는 작업은 작업순서에 따라 작업의 가공시간이 감소하며 (LPT order) 납기보다 늦게 완료되는 작업은 작업순서에 따라 작업의 가공시간이 증가할 때 (SPT order), 이 스케줄을 *V-shape* 이라 한다. ET모델에서 많은 종류의 목적함수들에 대하여 최적의 스케줄은 V-shape 임이 알려져 있다 [1]. 성질 1은 MAD/T_{max}의 문제에서도 최적 스케줄은 V-shape임을 나타낸다.

성질 1. 최적의 스케줄은 *V-shape*이다.

증명. [1]과 [7]에서 사용되어진 작업의 교환을 (pairwise interchange) 통한 증명을 이용하여 보일 수 있다. V-shape이 아닌 임의의 스케줄에서 두개의 작업을 골라 두 작업에 대해 V-shape의 형태가 되도록 만든 후 이 스케줄이 먼저 스케줄보다 목적함수의 값이 나쁘지 않음을 보이고 이와 같은 절차를 모든 작업들에 대해 V-shape 스케줄이 될 때까지 반복한다.

다음 세가지 성질들은 Hall [9]의 결과로부터 증명되어지며, MAD/T_{max} 문제의 복잡도를 보이기 위하여 사용된다.

성질 2. MAD/T_{max} 문제에서는 다음 중 하나의 스케줄이 존재한다.

a. $T^*_{max} = \Delta$를 만족하는 최적의 스케줄

b. 한 작업이 d에서 끝나는 최적의 스케줄

증명. $T^*_{max} < \Delta$를 만족하는 최적의 스케줄을 σ^*라고 하자. 만약 한 작업이 d에서 끝나면 증명이 된 것이다. 그렇지 않으면, d보다 먼저 가공을 시작하여 d보다 늦게 완료되는 작업을 α라고 하고 $0 < \varepsilon < \min\{d-(C_\alpha - p_\alpha), (C_\alpha - d), |T^*_{max} - \Delta|\}$라고 하자. Δ_E와 Δ_T를 전체의 작업들을 ε의 단위시간만큼 각각 더 앞으로 혹은 뒤로 옮겼을 때 목적함수값의 변화량이라고 정의하면 $\Delta_E = -\Delta_T$임을 쉽게 알 수 있다. 또, σ^*는 최적이므로 $\Delta_E, \Delta_T \le 0$를 만족하고 이는 $\Delta_E = \Delta_T = 0$를 나타낸다. 따라서 값의 변화없이 $T^*_{max} = \Delta$ 혹은 $C_\alpha = d + p_\alpha$를 만족할 때까지 모든 작업들을 뒤로 옮길수 있다.

σ^*를 최적 스케줄이라 하고 α를 σ^*에서 d에서 완료되거나 α를 작업의 가공중에 d가 존재하는 작업이라고 하자. $|E|$와 $|T|$를 각각 σ^*에서 작업 α보다 앞에 위치하는 작업의 수와 뒤에 위치하는 작업의 수라고 정의한다.

성질 3. $T^*_{max} < \Delta$이면, $|E|+1 \le |T|)+1$이다.

증명. 성질 2에 의해, $C_\alpha = d$ 이다. 모든 작업들을 $\varepsilon(0 < \varepsilon < 1)$의 단위 시간만큼 뒤로 옮겼을 때의 변화값은 $\varepsilon((|T|+1)-|E|) \ge 0$이다. 따라서 $|E| \le |T|+1$이다.

성질 4. $|E| \ge |T|-1$ 이다.

증명. 모든 작업들을 $\varepsilon(0 < \varepsilon < 1)$의 단위 시간만큼 앞으로 옮겼을 때의 변화값은 $\varepsilon((|T|+1) - |E|) \ge 0$이하이다. 따라서, $|E| \ge |T| - 1$이다.

성질 5. 최대 허용지연시간 Δ가 0이면 최적의 스케줄은 LPT순서이며 시작시간은 MS-d이고 완료시간은 d이다.

증명. 최대 허용지연시간이 0이므로 납기보다 지연되는 작업이 하나도 없다. 따라서, 모든 작업들의 완료시간은 d보다 작으며 이는 earliness의 절대편차의 합을 최소화하는 문제가 된다. 먼저 지연시간 (delay)을 삽입한 스케줄이 지연시간이 없는 스케줄보다 좋지 않다는 것은 같은 작업순서를 가지는 두개의 스케줄의 (하나는 지연시간이 있고 하나는 지연시간이 없는 스케줄) 목적함수를 비교함으로써 알 수 있다. LPT 순서는 두개의 작업을 교환하여 그 영향을 비교하는 방법으로 증명할 수 있다.

3. $\Delta \ge \theta$ 인 경우 MAD/T_{max} 문제

본 절에서는 $\Delta \ge \theta$ 인 경우 MAD/T_{max} 문제가 다항시간안에 해결된다는 것을 보인다.

Bagchi, Sullivan 그리고 Chang [1]은 MAD 문제가 $d \ge MS$-η 인 경우에는 비제약적 (unconstrained)이며 그렇지 않으면 제약적 (constrained)임을 보였다. 이를 MAD/T_{max} 문제에 적용하여 해를 구하기 위해 먼저 Δ의범위를 $\Delta \geq \eta$와 $\theta \leq \Delta < \eta$ 인 두 가지 경우로 나눈다.

$\Delta \geq \eta$ 경우에는 η 가 MAD문제의 모든 최적 스케줄에서 납기보다 늦은 작업들의 가공시간의 합을 포함할 수 있을 만큼 크다는 것을 나타낸다. 즉, MAD/T_{max}문제를 풀 때 MAD문제를 풀 때와 같은 방법을 사용하여도 얻어진 최적의 스케줄에 대한 T_{max}가 항상 Δ보다 작다. 따라서, 비제약적 MAD문제의 모든 최적 스케줄 (즉, 문헌 [1]에서의 모든 해)은 MAD/T_{max}문제의 최적 스케줄이 된다. 단, 목적함수 값은 모든 스케줄에 대하여 같지만 T_{max}값은 스케줄마다 다르나 제약조건은 만족한다.

$\theta \leq \Delta < \eta$ 인 경우 비제약적 MAD문제의 최적 스케줄을 고려해보자. B와 A를 유휴시간(idle time)이 없고 B의 마지막 작업의 완료시간과 A의 첫번째 작업의 시작 시간이 납기 d와 일치하도록 스케줄 되어있는 두개의 집합이라고 하자. Bagchi, Sullivan, 그리고 Chang [1]은 최적 스케줄은 (i) B의 첫번째에 가장 긴작업을 배치하고, (ii) 다음으로 긴 두개의 작업중 하나는 A의 마지막에 배치하고 또 다른 작업은 B의 두번째에 배치하고, (iii) 마지막으로 가장 짧은 작업은 B의 마지막에 배치하거나 A의 첫번째에 배치함으로 얻어질 수 있음을 보였다. 그들은 또한 최적의 스케줄의 갯수는 r이 다음과 같을 때 2^r이라는 것을 보였다.

$$r = \begin{cases} (n-1)/2, & n\text{이 홀수 일때} \\ n/2, & n\text{이 짝수 일때} \end{cases}$$

비제약적 MAD문제에서 Bagchi, Sullivan, 그리고 Chang [1]의 알고리즘을적용하여 구한 최적의 스케줄은 항상 $\theta \leq T_{max} \leq \eta$ 을 만족한다. 따라서, MAD/T_{max} 문제에서는 2^r개의 최적 스케줄 중에서 제약조건인 $T_{max} \leq \Delta$를 만족하는 스케줄을 찾아야 한다. 이는 MAD문제의 해결절차를 통해 구한 최적 스케줄의 T_{max}를 계산하면 된다. 또한, 해를 발견하는 과정에서 부분 스케줄의 T_{max}를 계산하여 $T_{max} \leq \Delta$인지를 확인함으로써 계산량을 줄일 수 있다. 즉, $T_{max} > \Delta$인 부분 스케줄은 더 고려할 필요가 없게 되므로 고려 대상에서 제외된다.

특히, $\Delta = \theta$ 인 경우에는 [1]의 두번째 과정에서 두개의 작업중 긴 것은 B에, 그리고 짧은 것은 A에 위치 시키고 한개의 작업이 남는 경우 B에 위치시키는 방법으로 유일한 최적 스케줄을 얻을 수 있다. 이 최적해는 $\theta \leq \Delta < \eta$ 인 경우에 항상 최적해중의 하나이며 최적해의 갯수는 Δ의 값에 따라 달라진다.

예제. 작업의 갯수가 5개이고 가공시간이 아래의 같이 주어졌을 때 MAD/T_{max} 문제를 고려해 보자.

〈표 1〉 MAD/T_{max} 문제

j	1	2	3	4	5
p_j	5	9	12	17	20

이 예에서 θ = p_1+p_3 = 5+12 = 17이고 η = p_2+p_4 = 9+17 = 26이다. 앞에서 설명한 것과 같이 MAD 문제의 최적 스케줄을 발견하는 과정을 이용하여 [그림 1]과 같은 스케줄을 얻을 수 있다. 먼저 $\Delta \geq \eta$ 경우를 생각해 보자. Δ가 26보다 큰 경우이므로 비제약적 MAD문

제의 모든 최적 스케줄의 갯수는 $2^r = 2^2 = 4$ 이고 이 스케줄들은 (5,4,2,1,3), (5,4,1,2,3), (5,3,2,1,4), (5,3,1,2,4)이며 이는 MAD/T_{max}문제에서도 최적의 스케줄이 된다. 단, 목적함수 값은 모든 스케줄에 대하여 같지만 각각의 스케줄마다 T_{max}값은 17, 21, 22, 26으로 다르며 제약조건인 $T_{max} \leq \Delta$은 만족한다.

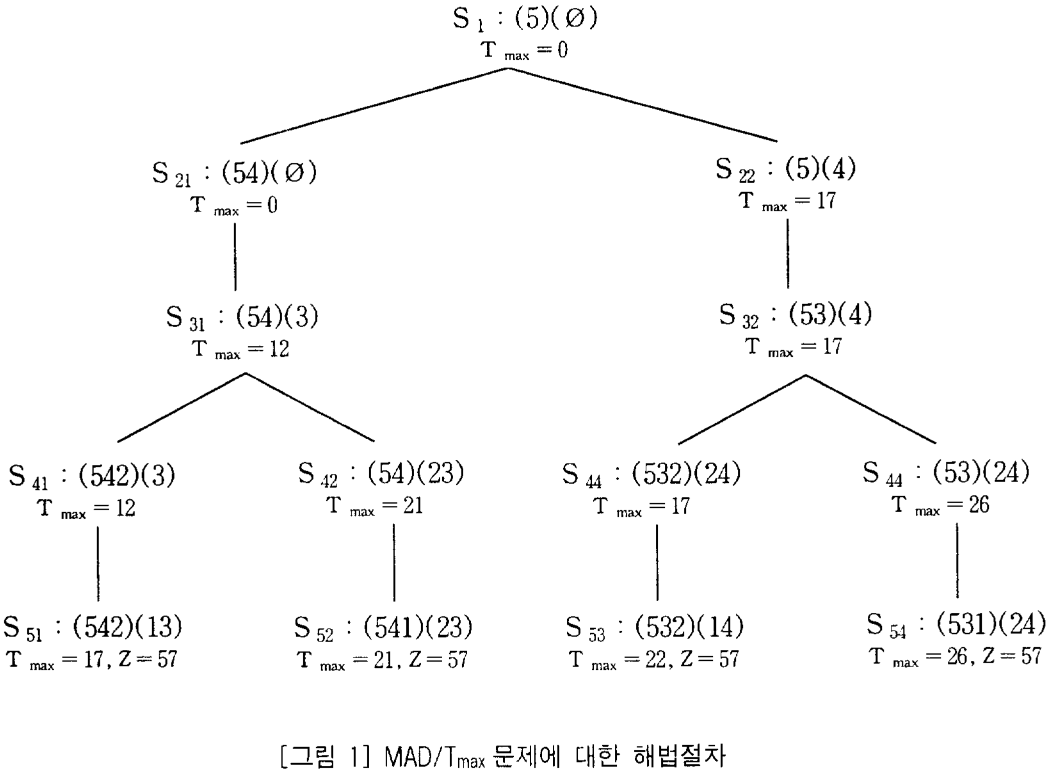
$\theta \leq \Delta < \eta$ 인 경우에 MAD/T_{max} 문제는 Δ의 값에 따라최적의 스케줄이 달라진다. MAD 문제를 해결하는 과정을 이용하여 얻어진 모든 스케줄에 대하여 T_{max}를 계산한다. 물론, 부분 스케줄에서 T_{max}값을 계산하고 이를 이용하여 스케줄 (5,3,1,2,4)는 생성하지 않을 수도 있다. 예제에서 $\Delta < 26$인 경우에는 스케줄 (5,4,2,1,3), (5,4,1,2,3), (5,3,2,1,4)이 최적해가 된다. 그러나, $\Delta < 21$인 경우에는 스케줄 (5,4,2,1,3)만이 제약조건을 만족하는 유일한 최적해이다.

4. $\Delta < \theta$ 인 경우 MAD/T_{max}문제

4.1. NP-complete의 증명

본 절에서는 $\Delta < \theta$ 인 경우 *MAD/*T_{max}가 NP-complete임을 증명한다. 다음과 같이 MAD와 *MAD/*T_{max}에 대응되는 결정 문제를 나타낼 수 있다.

(i) MAD : 공통의 납기 d가 주어질때 $C_j \geq$



[그림 1] MAD/T_{max} 문제에 대한 해법절차

p_j, j = 1,…, n를 만족하고 earliness와 tardiness의 합이 어떤 정수 y를 넘지 않는 단일기계에서의 스케줄이 존재하는가?

(ii) MAD/T_{max} : 공통의 납기 $\overline{d}$ 가 주어졌을때 $C_j \geq p_j$ (j=1,…, n)를 만족하고 earliness와 tardiness의 합이 어떤 정수 $\overline{y}$ 를 넘지 않고 동시에 tardiness의 최대값이 어떤 정수 Δ를 넘지 않는 단일기계에서의 스케줄이 존재하는가?

MAD/T_{max}가 NP class에 속해 있다는 것을 보이는 것은 간단하다. 이것이 NP-hard라는 것을 보이기 위해 우리는 MAD문제를 MAD/T_{max}문제로 축소 (reduce)한다. Hall, Kubiak 그리고 Sethi [5]는 NP-complete인 MAD문제의 예 (instance) P1을 보였다. 이들이 소개한 예 P1은 다음과 같다.

예 P1

$$p_{2i-1} = a_{2i-1} + B^i = a_{2i} + B^i, i=1, \cdots, n,$$

$$p_{2n+1} = B^{n+i}$$

$$p_{2n+1} = 2d, i=2, \cdots, 2n+3'$$

$$d = B + \sum_{i=1}^{n+1} + B^i$$

$$y = \sum_{i=1}^{n}(n+i+1)(p_{2i-1} + p_{2i}) + (2n+2)\sum_{i=1}^{2n+1} p_i + (2n+2)^2 d$$

단, $a_1, \cdots a_{2n}$은 $a_1 \leq \cdots \leq a_{2n}$이며

$B = \sum_{i=1}^{2n} a_i/2$를 만족하는 임의의 정수 n>0에 대하여 양의 정수를 나타내며, B^i는 B의 i 제곱을 나타낸다. 또, 예 P2가 다음과 같이 정의될때 $z(\sigma) \leq y$를 만족하는 예 P1의 스케줄 σ가 존재한다는 것이 $z(\sigma') \leq \overline{y}$와 $T_{max} \leq \Delta$를 만족하는 예 P2의 스케줄 σ'가 존재한다는 것과 필요충분 조건이 되도록 MAD/T_{max}의 예 P2를 구성한다.

예 P2

$$\overline{d} = \sum_{i=1}^{n} p_{2i} + \sum_{i=2n+1}^{4n+3} p_i,$$

$$\Delta = B + \sum_{i=1}^{n} B^i \text{and}$$

$$\overline{y} = y - \left\{(2n+2)\sum_{i=1}^{2n+1} p_i + (2n+2)^2 d\right\} + \left\{(2n+2)\left(\sum_{i=1}^{2n+1} p_i - \Delta\right) + 2d\sum_{i=1}^{2n+1} i\right\}$$

단, $p_1, \cdots, p_{4n+3}$과 d는예 P1에서와 같이 정의된다.

먼저 $z(\sigma)\leq y$를 만족하는 예 P1에 대한 최적의 스케줄 σ가 존재한다고 가정하자. $z(\sigma')\leq \bar{y}$와 $T_{max}\leq \varDelta$를 만족하는 P2에 대한 스케줄 σ'가 존재함을 보이면 된다. [5]에서 (i) 우선 작업 $2n$+1이 스케줄되고 시간 0에서 시작하며, (ii) 작업 2n+2, 2n+3,⋯, 4n+3은 끝에 연속적으로 스케줄되고, (iii) 한 작업을 d에 끝나며, 그리고 (iv) 작업 $2i$-1와 $2i$ 중의 하나는 $E\cup\{\alpha\}$에 속하는 σ는 i = 1, ⋯, n에 대해서 $\sum_{i\in E\cup\{\alpha\}}^{2n} a_i=\sum_{i\in T}^{2n} a_j=B$을 만족한다는 것을 보였다. [그림 2]에서와 같이 예 P2의 스케줄 σ'를 고려해보자. 우리는 다음과 같은 결과를 얻을수 있다.

$$
\begin{aligned}
z(\sigma') = {} & \bar{d}-(\bar{d}+\varDelta-\sum_{i=1}^{2n+1} p_i) \\
& +\bar{d}-(\bar{d}+\varDelta-\sum_{i=1}^{2n+1} p_i)+2d \\
& + \quad \vdots \\
& +\bar{d}-(\bar{d}+\varDelta-\sum_{i=1}^{2n+1} p_i+(4n+2)d\} \\
& +\sum_{i=1}^{n}(n-1+1)(p_{2i-1}+p_{2i}) \\
= {} & \bar{y}, \\
T_{\max} = {} & p_1+p_3+\cdots+p_{2n-1} \\
= {} & a_1+a_3+\cdots+a_{2n-1}+\sum_{i=1}^{n} B^i \\
= {} & \sum_{i\in E\cup\{\alpha\}}^{2n} a_i+\sum_{i=1}^{n} B^i \\
= {} & B+\sum_{i=1}^{n} B^i \\
= {} & \varDelta.
\end{aligned}
$$

따라서 σ'는 $z(\sigma')\leq \bar{y}$와 $T_{\max}=\varDelta$를 만족하는 예 P2의 스케줄이다.

4n+3	⋯	2n+1	2n	2n-2	⋯	2	1	3	⋯	2n-1

[그림 2] 보조정리 1에 대한 스케줄 σ'

반대로 σ'가 $z(\sigma')\leq \bar{y}$와 $T_{\max}=\varDelta$를 만족하는 예 P2에 대한 최적의 스케줄이라고 가정하자. σ'는 그림 1에서 보이는 것과 같이 $z(\sigma')\leq y$ 를 만족하는 예 P1에 대한 최적의 스케줄이 존재한다는 것을 나타낸다.

보조정리 1. 스케줄 에서 , (i) 작업 2n+2, 2n+3,⋯, 4n+3은 σ로부터 시작하여 연속적으로 스케줄되며 (ii) $T^*_{\max}=\Delta$이다.

증명. 모순임을 보이기 위해 σ'은 (i)를 만족하지 않는다고 가정하자. 성질 1에 의해 2n+2, 2n+3,⋯, 4n+3중 하나는 마지막에 수행되어야 하며 다른 모든 작업들은 LPT의 순으로 수행된다. 따라서,

$$z(\sigma')= (4n+i+2)\sum_{i=1}^{2n+1} p_i+(4n+2)(2d-\Delta) + 2d\sum_{i=1}^{2n} + \Delta.$$

$2d$를 $z(\sigma')$에서의 $\sum_{i=1}^{2n+1} p_i + B^{n+1}$로 변경시키면 다음과 같은 결과를 얻을 수 있다.

$$\begin{aligned} z(\sigma')-\bar{y} &> (2n+1)\sum_{i=1}^{2n+1} p_i+2n(\sum_{i=1}^{2n+1} p_i-\Delta) \\ &\quad +(4n+2)B^{n+1}-(2n+1)\sum_{i=1}^{2n+1}+B^{n+1}) \\ &> 0. \end{aligned}$$

이는 σ'가 최적이라는 사실에 모순이다.

σ'는 (ii)를 만족하지 않는다고 가정하자. 그러면 σ'에서 $|T| \le 2n$ 이고 $|E| \ge 2n+2$이다. 이는 성질 3에 대해 모순이다.

작업 2n+2, 2n+3,⋯, 4n+3이 σ'의 처음에 연속적으로 스케줄되고 σ'에서 $T^*_{\max}=\Delta$를 만족하므로(보조정리 1) 작업 1, 2,⋯, 2n+1에 대하여 $\bar{d}+\Delta$에 끝나는 최적 스케줄을 찾으면 된다. 이는 $\Delta= p_1+ p_3+ \cdots + p_{2n-1}$이기 때문에 $\Delta=\eta$를 만족하는 MAD/T_{max} 문제이다. 3절에서 언급한 바와 같이 이 경우의 유일한 최적의 스케줄은 작업 2가 $\bar{d}$에 끝나고 작업 2n-1가 $\bar{d}+\Delta$에 끝나고 작업 2n+1, 2n, 2n-2,⋯, 2는 LPT의 순으로 $\bar{d}$ 이전에 수행되며,작업 1, 3, ⋯, 2n-1는 SPT의 순으로 $\bar{d}$ 이후에 수행된다. 따라서 최적 스케줄 σ'는 [그림 2]와 같이 스케줄된다.

위에서 설명된 MAD 문제에서 MAD/T_{max} 문제로 변환하는 과정이 다항시간 내에 해결된다는 것은 쉽게 보일수 있으므로 다음의 정리를 얻을 수 있다.

정리 1. *MAD/*T_{max}문제는 *NP-complete* 이다.

4.2. 자기발견적 알고리즘의 개발

MAD/T_{max}문제는 NP-complete이기 때문에 자기발견적 알고리즘을 이용하여 이 문제를 해결하는 것이 바람직하다. 그러므로, $\Delta < \theta$ 인 경우에 MAD/T_{max}문제에 대한 자기발견적 알고리즘을 제시하고자 한다. 본 논문에서 제안된 알고리즘은 스케줄을 생성하는 절차와 해를 향상시키는 절차로 구성되어져 있다. 최대 허용 지연시간 Δ가 매우 작은 MAD/T_{max}문제에 대한 스케줄을 생성하기 위해서 다음과 같은 개념이 고려되었다: (1) 최적의 스케쥴은 V-shape 이다, (2) 작업사이의 유휴시간 (idle time)이 존재하지 않는다, (3) 최대 허용지연시간에 대한 제약이 존재한다, (4) 모든 스케줄이 항상 $d+\Delta$인 시점에서 끝나진 않는다. 스케줄을 생성하는 절차는 Sundararaghavan과 Ahmed [8]의 MAD 알고리즘에 기초하여 개발되었으며 스케줄을 생성한 후 얻어진 해를 향상시키기 위해 다음의 세가지 성질을 제시한다. 두개의 작업을 교환하거나 전체 스케줄을 이동하는 방법으로 목적함수의 값이 감소하는 경우를 찾아 내었다.

성질 6. 주어진 스케줄 S에 대하여 (i) 납기에 걸쳐진 작업 x가 존재하고 $|E| < |T| + 1$이면 $C_x = d$를 만족할 때까지 스케줄 S를 왼쪽으로 이동시킴으로서 더 좋은 해를 얻을 수 있다. (ii) 납기에 걸쳐진 작업 x가 존재하지 않고 $|E| < |T|$이면 $|E| \geq |T|$이고 $C_{[|E|+1]} = d$일 때까지 왼쪽으로 이동시킴으로서 더 좋은 해를 얻을 수 있다.

증명. (i) 스케줄 S를 $C_x = d$을 만족할때까지 왼쪽으로 이동시킨 스케줄을 S'이라 하고 변화량을 계산해 보면 다음과 같다.

$$Z(S') - Z(S) =$$

$$(C_x - d)|E| - (C_x - d)(|T| + 1)$$

$$= -(C_x - d)(|T| + 1 - |E|)$$

$$< 0 \quad (\because |T| + 1 > |E|).$$

따라서, $Z(S) > Z(S')$.

(ii) (i)과 같은 방법으로 증명 가능.

성질 7. S를 납기에 걸친 작업이 존재하는 스케줄이라고 가정하자. (i)두개의 작업 $E_{[i]}$와 $T_{[j]}$가 $i > j$이고 $0 < p_{T[j]} - p_{E[i]} \leq \alpha$이면 두개의 작업 $E_{[i]}$과 $T_{[j]}$를 교환하여 더 좋은 해를 얻을 수 있다. (ii) 만약 두개의 작업 $E_{[i]}$와 $T_{[j]}$이 $i < j$이고 $0 < p_{E[i]} - p_{T[j]} \leq (C_x - d)$를 만족하면 두개의 작업 $E_{[i]}$과 $T_{[j]}$를 교환하여 더 좋은 해를 얻을 수 있다.

증명.

(i) $\delta(<\alpha) = p_{E[i]} - p_{T[j]}$라고 정의하면,

$$Z(S') - Z(S) = -i\delta + j\delta$$

$$= \delta(j - i) < 0 \ (\because i > j).$$

따라서, $Z(S) > Z(S')$

(ii)(i)와 같이 증명 가능.

성질 8. $C_x = d$를 만족하는 작업 x 가 존재하

는 스케줄 S에 대해 생각해 보자. (i) 두개의 작업 $E_{[i]}$와 $T_{[j]}$가 $i > j+1$이고 $0 < p_{T[j]} - p_{E[i]} \le (d = C_x + p_x)$이면 두개의 작업 $E_{[i]}$과 $T_{[j]}$를 교환하여 더 좋은 해를 얻을 수 있다. (ii) 만약 두개의 작업 $E_{[i]}$와 $T_{[j]}$이 $i+1 < j$이고 $0 < p_{E[i]} - p_{T[j]} (d = C_x - d)$를 만족하면 두개의 작업 $E_{[i]}$과 $T_{[j]}$를 교환하여 더 좋은 해를 얻을 수 있다.

증명. (i), (ii) 성질 7에서의 (i)와 같은 방법으로 증명할 수 있다.

자기발견적 알고리즘

단계 1. 초기화.

$i = n, R = \Delta, L = MS - \Delta,$

$R_{set} = 0,\ L_{set} = 0.$

단계 2. 작업 삽입.

2-1. $R \ge L$이면

i 번째 작업을 R_{set}에 삽입한다.

$R \leftarrow R - p_i.$

2-2. $R < L$이면

i 번째 작업을 L_{set}에 삽입한다.

$L \leftarrow L - p_i.$

단계 3 $i=1$이면, 단계 4로 이동.

그렇지 않으면 $i \leftarrow i-1$ 단계 2로 이동

단계 4. L_{set}에 있는 작업들은 LPT의 순서로, R_{set}에 존재하는 작업들은 SPT의 순서로 나열하고 두개의 나열된 열을 연결하여 $C_{[n]} = d + \Delta$를 만족하는 스케줄을 완성한다.

단계 5. 얻은 스케줄에 대한 역방향의 (reversed)스케줄을 찾는다. 각각의 스케줄에 대하여 성질 6부터 8까지 적용시킨다. 만약 성질 7이나 8이 적용되면 단계 5를 반복한다.

단계 6. 두개의 스케줄에 대한 목적함수 값을 비교하여 더 좋은 값을 갖는 스케줄을 선택한다.

제안된 알고리즘은 크게 (1) 스케줄의 생성과정 (단계 1 - 단계 4) (2) 해의 향상 과정 (단계 5 - 단계 6)으로 나눌 수 있다. 단계 5에서는 생성된 스케줄의 역방향의 스케줄을 생성하기 때문에 두개의 스케줄이 동시에 고려대상이 된다. 스케줄에 대한 이동과 (성질 6) 교환이 (성질 7, 성질 8)이 각 스케줄에 적용된다.

4.3. 실험 결과

제안된 알고리즘의 수행정도를 보기 위해 Sundararaghavan과 Ahmed [9]의 논문에서 제시된 8개의 문제를 선택하였다. 제안된 알고리즘은 Borland C++로 프로그래밍하였으며 Pentium 60에서 실행 되었다. 각 문제에 대해서는 $1/2 \sum_{j \in N} p_j$보다 작은 2개의 최대 허용 만기값을 생성하였다.

<표 2>에서 Δ값은 각각

$$\frac{1}{2} \sum_{j \in N} p_j - p_n - \frac{k(p_n - p_1)}{n}$$ 와

$$\frac{1}{2} \sum_{j \in N} p_j - p_n + \frac{k'(p_n - p_1)}{n}$$ 에 의하여 계산된 값을 이용하였다.

공통 납기는 문제에 제약을 가하지 않도록 크게 하기 위해 모든 작업들의 가공시간의 합으로 (makespan) 설정하였다. 각 예제에 대하여

제안된 알고리즘과 Sundararaghavan과 Ahmed의 [8] 알고리즘을 적용하여 얻은 결과는 〈표 2〉에 나타나 있다. $\Delta \le \sum_{j\in N} p_j - p_n$일 경우, 두 알고리즘 모두 좋은 결과를 나타내는 반면에 $\frac{1}{2}\sum_{j\in N} p_j - p_n < \Delta < \frac{1}{2}\sum_{j\in N} p_j$일 경우는 제안된 알고리즘이 훨씬 좋은 결과를 나타내고 있다. 이 경우 스케줄의 이동 과정과 작업들의 교환 과정이 해의 값을 향상시켰음을 알 수 있다.

5. 결 론

본 논문에서는 주어진 최대 허용 지연시간(maximum allowable tardiness) 이내에 평균절대 편차를 최소화하는 MAD/Tmax문제를 다루었다. MAD문제를 다룬 이전의 결과들로부터 MAD/Tmax 문제에 대한 몇가지 성질을 발견하였고 MAD/Tmax문제가 $\Delta < \theta$인 경우 NP-complete임 보였다. 이 경우 문제를 해결하기 위하여 자기발견적 알고리즘을 제안하였다. 제안된 알고리즘의 수행도를 측정하기 위하

〈표 2〉 Computational results when $\Delta \le \sum_{j\in N} p_i - p_n$ or $\frac{1}{2}\sum_{j\in N} p_i - p_n < \Delta < \frac{1}{2}\sum_{j\in N} p_i$

No. of Jobs	Processing Times	Due Date	Δ	S & A Heuristic	nZ(S)	Proposed Heuristic	nZ(S)
6	1,10,11,48,50,53	197	k=1	6 5 4 1 2 3	182	6 5 4 1 2 3	180
			k'=n	6 3 2 1 4 5	202	6 4 1 2 3 5	155
6	22,25,36,65,73,84	305	k=1	6 5 4 2 1 3	355	6 5 4 2 1 3	355
			k'=n	6 4 2 1 3 5	323	6 4 2 1 3 5	323
6	3,37,57,75,81,99	352	k=1	6 5 4 2 1 3	405	6 5 4 2 1 3	405
			k'=n	6 3 2 1 4 5	399	6 4 2 1 3 5	371
6	7,10,11,69,77,92	266	k=1	6 5 4 1 2 3	264	6 5 4 1 2 3	264
			k'=n	6 3 2 1 4 5	307	5 4 1 2 3 6	227
9	3,6,7,8,14, 30,45,48,72	233	k=1	9 8 7 5 4 3 1 2 6	301	9 8 7 5 4 3 1 2 6	301
			k'=n	9 6 4 3 1 2 5 7 8	284	9 6 4 3 1 2 5 7 8	283
14	1,2,4,5,8,23,31, 53,55,65,68,69,90,92	566	k=1	14 13 12 10 8 5 3 1 2 4 6 8 10 12	1099	14 13 12 10 8 5 3 1 2 4 6 7 9 11	1099
			k'=n	14 13 11 9 7 5 3 1 2 4 6 7 9 11	1073	14 13 11 9 7 5 3 1 2 4 6 8 10 12	1073
14	4,13,14,21,33,37,38, 48,63,78,80,85,93,94	701	k=1	14 13 12 10 7 6 3 2 1 4 5 8 9 11	1613	14 13 12 10 7 6 3 2 1 4 5 8 9 11	1613
			k'=n	14 12 10 7 5 4 1 2 3 6 8 9 11 13	1603	14 12 10 7 5 4 1 2 3 6 8 9 11 13	1603
14	8,21,23,29,35,36,44, 51,52,62,69,81,84,85	680	k=1	14 13 12 9 8 6 4 1 2 3 4 7 10 11	1724	14 13 12 9 8 6 4 1 2 3 4 7 10 11	1724
			k'=n	14 12 10 7 5 4 1 2 3 6 8 9 11 13	1717	14 12 10 7 6 4 1 2 3 5 8 9 11 13	1716

* : a schedule that ends $d+\Delta$

여 8개의 예제를 문헌에서 선택한 후 이의 계산 결과를 제시하였다. 또한, 최대 허용 지연시간의 범위에 따라 ($\Delta \geq \theta$인 경우) MAD/Tmax문제가 다항시간안에 해결이 가능한 특별한 경우를 보였다. 앞으로의 연구는 ET 모델에서 다목적인 경우에 efficient한 해들을 얻을 수 있는 알고리즘의 개발, 준비시간이 다른 작업들의 MAD문제를 해결하는 효과적인 자기발견적 알고리즘의 개발이 될 것이다.

참 고 문 헌

[1] Bagchi, U., R. S. Sullivan and Y. L. Chang, "Minimiznig Mean Absolute Deviation of Completion Times about a Common Due Date," *Naval Research Logistics Quarterly*, Vol. 33(1986), pp. 227-240.

[2] Bagchi, U., R. S. Sullivan and Y. L. Chang, "Minimizing Absolute and Squared Deviation of Completion Times with Different Earliness and Tardiness Penalties and a Common Due Date," *Naval Research Logostics*, Vol. 34(1987), pp. 739-751.

[3] K.R. Baker, G.D. Scudder,"Sequencing with earliness and tardiness penalties : A Review, Operations Res.", Vol. 38, No. 1(1990), pp.22-36.

[4] M. Garey, R. Tarjan and G. Wilfong, "One-Processor Scheduling with SymmetricEarliness and Tardiness Penalties," *Math. Ops Res.*, Vol. 10(1988), pp.330-348.

[5] Hall, N. G., W. Kubiak and S. P. Sethi, "Earliness-Tardiness Scheduling problems, II : Deviation of Completion Times about a Restrictive Common Due Date," *Operation Research*, Vol. 39, 5(1991), pp. 847-856.

[6] Kanet, J.J., "Minimizing the Average Deviation of Job Completion times about a Common Due Date," *Naval Research Logitics Quarterly*, Vol. 28(1981), pp. 643-651.

[7] Raghavachari, M. "A V-Shape Property of Optimal Schedule of Jobs About a Common Due Date," *Eur. J. Opnl.* Res. 23(1986), pp.401-402.

[8] Sundararaghavan, P. S. and M. U. Ahmed, "Minimizing the Sum of Absolute Lateness in Single-Machine and Multimachine Scheduling," *Naval Research Logitics Quarterly*, Vol. 31(1984), pp.325-333.