최대 지연시간을 고려한 Earliness/Tardiness 모델에서의 스케줄링

서 종 화, 이 동 훈
충남 연기군 조치원읍 서창동 208번지 고려대학교 자연과학대학 전산학과
김 채 복
충북 청원군 강내면 한국교원대학교 기술교육학과

Abstract

We consider a nonpreemptive single-machine scheduling problem to minimize the mean squared deviation(MSD) of job completion times about a common due date d with a maximum tardiness constraint, i.e., maximum tardiness is less than or equal to the given allowable amount, Δ(MSD/T_{max} problem). We classify the Δ-unconstrained, Δ-constrained and tightly Δ-constrained cases in the MSD/T_{max} problem. We provide bounds to discern each case for the problem. It is also shown that the Δ-unconstrained MSD/T_{max} problem is equivalent to the unconstrained MSD problem and the tightly Δ-constrained MSD/T_{max} problem with n jobs and a maximum allowable tardiness Δ can be converted into the constrained MSD problem with a common due date Δ and $n-1$ jobs. Finally, the solution procedure for MSD/T_{max} problem is provided.

1. 서론

수십 년간 스케줄링에 관한 연구는 "regular measure"에 초점을 맞추어 왔으나, "just-in-time"의 개념이 도입된 이래로, earliness와 tardiness를 함께 최소화시키는 "nonregular measure" 에 관한 많은 연구가 진행되고 있다. Earliness와 tardiness를 함께 최소화시키는 문제는 Merten과 Muller [5], Sidney [6], Kanet [4] 등에 의해 다루어졌다. Baker와 Scudder [2]는 earliness와 tardiness를 함께 최소화시키는 문제와 관련해 많은 종류의 목적함수를 조사하였다. 본 논문에서는 공통납기를 중심으로 작업 종료시간의 평균 제곱편차(MSD)를 최소화시키는 것을 목적함수로 한다.

목적함수 MSD와 관련해 Bagchi와 그의 동료들 [1]은 주어진 납기가 충분히 여유가 있을 때(unconstrained)와 그렇지 않을 때(constrained)의 경우로 분류하여 각 경우에 관한 알고리즘을 제안하였고 De와 그의 동료들 [3]은 [1]의 논문의 논리적 결함을 수정하고 MSD문제를 tightly constrained와 constrained 그리고 unconstrained인 경우로 나누는 下限(lower bound)과 上限(upper bound)을 규명하였다.

본 논문에서는 공통납기가 makespan 보다 크거나 같다는 가정 하에 최대 허용 지연시간(maximum allowable tardiness)이 제약조건으로 주어졌을 경우 각 작업의 종료시간의 평균 제곱편차를 최소화하는 스케줄을 발견하는 문제(MSD/T_{max})를 다룬다.

다음절에서 MSD/T_{max} 문제를 정의하고 최대 허용 지연시간(Δ)에 따라 Δ-unconstrained와 Δ-constrained 그리고 tightly Δ-constrained 경우로 분류하며, 문제 몇가지 성질들에 대하여 기술한다. 3절에서는 Δ-unconstrained MSD/ T_{max} 문제는 unconstrained MSD 문제와 동일함을 보이고 Δ-unconstrained 경우와 Δ-constrained 경우를 분리하는 Δ 값의 上限을 보인다. 4절에서는 n개의 작업을 가지고 최대 허용 지연시간이 Δ인 tightly Δ-constrained MSD/T_{max} 문제가 $n-1$개의 작업을 가지고 공통납기가 Δ인 MSD 문제로 변환될 수 있음을 보이고 Δ-constrained 경우와 tightly Δ-constrained 경우를 분리하는 Δ 값의 下限을 밝힌다. 마지막으로, MSD/T_{max} 문제의 최적해를 찾는 알고리즘을 제시한다.

2. MSD/T_{max} 문제

수행될 작업들이 n개인 nonpreemptive 단일 프로세서 스케줄링 문제를 고려해 보자. 작업 j는 처리시간(processing time)이 p_j 이고, 모든 작업들이 공통납기 d를 가지고 있다고 하자. 또한 오직 최대로 허용된 지연시간이 해에 미치는 영향을 분석하기 위하여 공통납기 d는 모든 작업들의 처리시간의 합보다 적지 않다고 가정하자. MSD/T_{max} 문제는 최대 지연시간이 최대 허용 지연시간 보다 크지 않고 공통납기를 기준으로 하는 각 작업 종료시간의 평균 제곱편차를 최소화하는 스케줄을 찾는 문제이다. 본 논문에서 사용되는 표기들을 다음과 같이 정의하자.

Δ = 최대 허용 지연시간
p_j = 작업 j의 처리시간
C_j = 작업 j의 종료시간
R_j = 작업 j의 시작시간
$Z(S)$ = 스케줄 S의 목적함수 값
$\overline{C} = (1/n)\Sigma_{i=1}^{n} C_i$
$MS = \Sigma_{i=1}^{n} p_i$, and
$R_s = d + \Delta - MS$

위의 정의들을 사용하여 MSD/T_{max} 문제를 수학적으로 정의하면 다음과 같다.

$$\min\ Z(S) = \frac{1}{n}\sum_{j=1}^{n}(C_j - d)^2$$
$$\text{subject to}\quad T_{\max} \le \Delta$$

정의. *[1]에 나타난 정리 7과 명제 2와 4 그리고 5를 만족하는 스케줄을 v-about-d라 한다.*

다음의 두가지 성질들은 스케줄이 MSD/T_{max} 문제의 최적해가 되기 위한 필요조건이다.

성질 1. *MSD/T_{max} 문제의 최적 스케줄은 v-about -d 이다.*

증 명. 위의 성질은 "작업 교환 argument"를 사용하여 쉽게 증명될 수 있다. ♣

성질 2. *MSD/T_{max} 문제의 최적 스케줄에서는 가장 큰 작업이 가장 먼저 처리된다.*

증 명. 모순을 보이기 위하여 가장 먼저 처리된 작업이 가장 큰 작업(즉, 작업 n)이 아닌 최적 스케줄 S가 존재한다고 가정하자. 성질 1에 의하여 작업 n은 가장 나중에 처리되어야 한다. S에서 처음으로 처리된 작업과 작업 n을 바꾸고 작업 n의 종료시간을 S의 처음 작업의 종료시간과 일치시킨 스케줄을 S' 이라고 하면 S'의 목적함수 값이 S의 값보다 작음을 쉽게 보일 수 있으며 이는 모순을 보이는 것이다. ♣

최대 허용 지연시간이 Δ이고 공통납기가 d인 MSD/T_{max} 문제를 고려해 보자. 만약 [1]이 제안한 알고리즘으로 구한 최적해가 최대 허용 지연시간 제약조건을 만족할 경우 이 문제를 Δ-unconstrained MSD/T_{max} 문제라고 정의하고, 그렇지 않을 경우에는 Δ-constrained MSD/T_{max}로 정의한다.

기존의 constrained MSD 문제에서는 [3]에서 지적한대로 최적인 스케줄이 시간 0에서 시작되지 않는 경우가 발생하는데, Δ-constrained 문제에서도 이와 비슷하게 최적해의 최대 지연시간이 최대 허용 지연시간보다 작은 경우가 발생한다. 다음의 예를 보자.

작업 j	1	2	3	4	5
처리시간 p_j	5	300	800	900	1000

표 1. d = 4000이고 Δ = 944인 MSD/T_{max} 문제

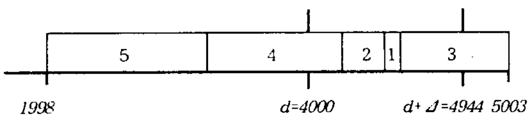

그림 1. [1]이 제안한 알고리즘으로 얻은 최적해

그림 1에서 보듯이, [1]이 제안한 알고리즘에 의해 구한 최적해가 최대 허용 지연시간의 제약조건을 만족하지 않음을 알 수 있다. 따라서 표 1에 제시된 문제는 Δ-constrained MSD/T_{max} 문제이다. 표 2는 성질 1과 성질 2를 만족하면서 최대 지연시간이 최대 허용 지연시간과 동일한 모든 스케줄과 목적함수 값을 나타내고 있다. 그림 2와 같이 C_3=4943 인 스케줄 (5, 4, 2, 1, 3)을 생각해 보자. 이 스케줄의 목적함수 값(416566)은 표 2에 나타난 스케줄들의 목적함수 값보다 작으므로, MSD/T_{max} 문제의 최적 스케줄은 반드시 최대 지연시간이 최대 허용 지연시간보다 작음을 알 수 있다.

스케줄	MSD 값
(5, 3, 2, 1, 4)	417687
(5, 4, 2, 1, 3)	416567

표 2. 성질 1과 성질 2를 만족하면서 최대 지연시간이 최대 허용 지연시간과 동일한 스케줄들의 MSD 값

Δ-constrained MSD/T_{max} 문제에서, 주어진 최대 허용 지연시간이 어떤 값보다 작으면 최적 스케줄의 최대 지연시간은 반드시 최대 허용 지연시간과 동일해야 하는데, 이러한 MSD/T_{max} 문제를 특별히 tightly Δ-constrained MSD/T_{max} 문제라고 정의한다. 우리는 4절에서 어떤 MSD/T_{max} 문제가 tightly Δ-constrained 경우가 되기 위한 충분조건을 제시할 것이다.

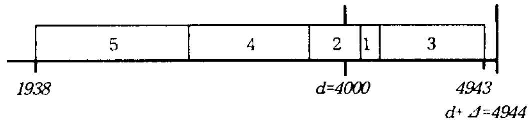

그림 2. 최대 지연시간이 최대 허용지연시간보다 작은 스케줄

3. Δ-unconstrained MSD/T_{max} 문제

최대 허용 지연시간이 충분히 큰 Δ-unconstrained MSD/T_{max} 문제는 정리 1에서 보이는 것처럼 기존의 unconstrained MSD 문제와 동일하다.

정리 1. *Δ-unconstrained MSD/T_{max} 문제는 unconstrained MSD 문제와 동일하다.*

증 명. Δ-unconstrained MSD/T_{max} 문제의 정의에 의해서 최대 허용 지연시간은 최적 스케줄을 찾는데 아무런 제약조건이 되지 않는다. 따라서, 위의 정리는 자명하다. ♣

[1]에서 증명되었듯이, unconstrained MSD 문제는 작업 종료시간들의 분산을 최소화시키는 문제(CTV 문제)와 동일하므로, Δ-unconstrained MSD/T_{max} 문제도 CTV 문제와 동일하다. 다음의 보조 정리와 정리는 주어진 MSD/T_{max} 문제가 Δ-unconstrained 경우이기 위한 충분조건을 제시한다.

보조 정리 1. *$\overline{C_l}=(1/2)(MS+p_n)+R_s$라 정의하면 최대 허용 지연시간의 제약조건을 만족하면서 평균 종료시간이 $\overline{C_l}$ 보다 크거나 같은 스케줄 중에는 CTV를 최소화하는 스케줄이 존재한다.*

증 명. CTV를 최소화하는 스케줄 S가 $(n, i_2, \ldots, i_n)$라 가정하자. 단, i_j는 j번째 위치에 있는 작업의 index를 나타낸다고 하자. 그렇다면 이 스케줄의 CTV 값과 동일한 CTV 값을 갖는 dual 스케줄 S'이 존재하고 $(n, i_n, \ldots, i_2)$의 형태를 가진다. 이 두 개의 스케줄이 모두 R_s에서 시작한다고 가정하자(CTV 값은 스케줄의 시작 시간과는 무관하다). 그러면, 이 스케줄들이 가지는 작업 평균 종료시간은 다음과 같다.

$$\overline{C_s}=\frac{1}{n}\{np_n+(n-1)p_{i_2}+\ldots+p_{i_n}\}+R_s$$

$$\overline{C_{s'}}=\frac{1}{n}\{np_n+(n-1)p_{i_n}+\ldots+p_{i_2}\}+R_s$$

위의 두 식을 더하면, $(\overline{C_s}+\overline{C_{s'}})=MS+p_n+2R_s$ 이 된다. $\overline{C_s}$와 $\overline{C_{s'}}$의 평균을 $\overline{C_l}$이라 하면, $\overline{C_l}\leq\max\{\overline{C_s},\overline{C_{s'}}\}$ 이 된다. ♣

정리 2. *최대 허용 지연시간이 $(1/2)(MS-p_n)$보다 크거나 같은 MSD/T_{max} 문제는 Δ-unconstrained 이다.*

증 명. 보조 정리 1에 의해 만약 주어진 공통납기 d가 $\overline{C_l}$보다 작거나 같으면, MSD/T_{max} 문제의 최적 해는 CTV 값을 최소화하는 스케줄을 찾아 이 스케줄이 가지는 작업들의 평균 종료시간을 d와 일치시킴으로써 구할 수 있다. 여기서, 조건 $d \le \overline{C_l}$은 다음에서 보이는 바와 같이 $\Delta \ge (1/2)(MS-p_n)$으로 나타내어 질 수 있다.

$$d \le \overline{C_l} = \frac{1}{2}(MS+p_n)+d+\Delta-MS,$$

$$0 \le \Delta - \frac{1}{2}(MS-p_n),$$

$$\Delta \le \frac{1}{2}(MS-p_n). \qquad ♣$$

4. Δ-constrained MSD/T_{max} 문제

다음의 정리는 n개의 작업을 가지며 주어진 최대 허용 지연시간이 Δ인 tightly Δ-constrained MSD/T_{max} 문제가 $n-1$개의 작업을 가지며 공통납기가 d인 MSD 문제로 변환될 수 있음을 보인다.

정리 3. *최대 허용 지연시간이 Δ 이고 작업 1, ..., n 을 가진 tightly Δ-constrained MSD/T_{max} 문제는 공통납기가 Δ이고 작업 1, ..., n-1을 가지는 constrained MSD 문제로 변환될 수 있다.*

증 명. 스케줄 $S=(i_1,\ldots,i_n)$를 tightly Δ-constrained MSD/T_{max} 문제의 최적해라고 하자. 성질 2에 의해 i_1은 처리시간이 가장 큰 작업인 n이다. 정리 3을 증명하기 위하여 스케줄 $\overline{S}=(i_n,i_{n-1},\ldots i_2)$은 공통납기가 Δ이고 작업 1, ..., n-1을 스케줄 하는 constrained MSD 문제의 최적해임을 보인다. 스케줄 S와 S'은 각각 그림 3(a)와 3(b)에서 보여지고 있다.

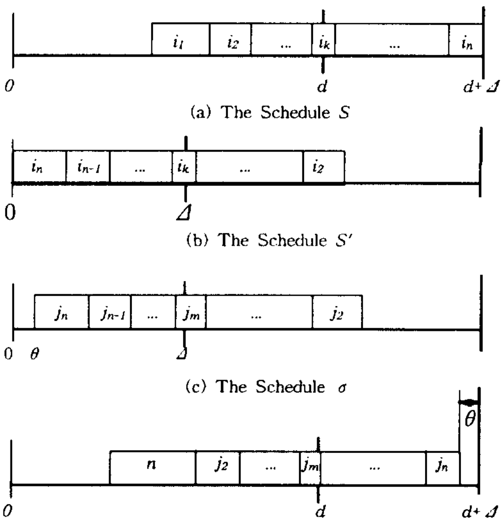

그림 3. tightly Δ-constrained MSD/T_{max} 문제와 Δ-constrained MSD 문제의 관계

모순을 보이기 위하여 $\overline{S}$가 constrained MSD 문제의 최적해가 아니라고 가정하자. $\overline{\sigma}=(j_n,j_{n-1},\ldots,j_2)$가 $Z(\overline{\sigma})<Z(\overline{S})$인 constrained MSD 문제의 최적해라고 하자. j_n의 시작시간이 $\theta(\ge 0)$라고 가정하자. 스케줄 σ는 다음과 같이 얻어진다: (1) 스케줄 $\overline{\sigma}$의 순서를 역으로 하고, (2) 작업 j_n을 처음에 놓은 다음, (3) 작업 j_n의 종료시간을 $(d+\Delta-\theta)$로 만든다. 스케줄 σ와 $\overline{\sigma}$는 각각 그림 3(c)와 3(d)에서 보여지고 있다. 스케줄 σ는 작업 1, 2, ..., n을 스케줄 하는 MSD/T_{max}의 해이다. 네 가지 스케줄들의 목적함수 값들의 관계는 다음과 같다.

$$Z(\sigma) = Z(\overline{\sigma})+(\Delta-\theta)^2$$

$$< Z(\overline{S})+\Delta^2 = Z(S).$$

이는 S가 최적해라는 사실에 모순이 되며 따라서 $\overline{S}$는 constrained MSD 문제의 최적 해이다. ♣

다음의 보조 정리 2와 정리 4에서는 MSD/T_{max} 문제가 tightly Δ-constrained MSD 문제이기 위한 충분조건을 제시한다.

보조 정리 2. *δ가 $d+\Delta$에 끝나고 큰 처리 시간순(LPT order)으로 처리된 스케줄의 평균 종료시간이라 하자. 즉, $R_s=d+\Delta-MS$ 이면 $\delta=(1/n)\{np_n+(n-1)p_{n-1}+\ldots+p_1\}+R_s$이다. 또한, $k=\max\{i \mid \Sigma_{j=i}^{n}p_j+R_s>\delta\}$이고, 스케줄 $S=(n, n-1,\ldots,k,k-1,1,\ldots,k-2)$ 에서 $C_{k-2}=d+\Delta$ 일 때의 평균 종료시간을 $\overline{C_u}$라 하자. 만약 주어진 공통납기가 δ보다 작거나 같으면, 어떤 MSD/T_{max} 문제의 가능한 최적해의 평균 종료시간도 $\overline{C_u}$보다 작다.*

증 명. 스케줄 S는 $R_{k-1}>\delta\ge d$ 이기 때문에 v-about-d 가 아니고 따라서 MSD/T_{max} 문제의 최적해가 될 수 없다. 이 보조 정리를 증명하기 위해 스케줄 S를 v-about-d로 만들려고 하면 스케줄의 평균 종료시간이 감소된다는 것을 보이면 된다. S를 v-about-d로 만드는 방법은 두 가지가 있다. 한 가지는 S를 $R_{k-1}<d$ 가 될 때까지 왼쪽으로 이동시키는 것이다. 그러나 이러한 방법은 각 작업의 종료시간을 감소시키며 따라서 평균 종료시간도 감소된다.

S를 v-about-d로 만드는 다른 방법은 작업 1을 왼쪽으로 이동시키기 위하여 작업 1의 전후에 있는 작업들을 서로 바꾸는 것이다. 이러한 방법도 결국 스케줄의 평균 종료시간을 감소시킴을 보이기 위하여 작업 1 앞에 스케줄 된 작업 e와 작업 1 뒤에 스케줄 된 작업들 $i_1,i_2,\ldots,i_t(0\le t\le k-2)$을 교환한 스케줄(그림 4)을 고려해 보자. (작업 1 앞에 있는 작업을 스케줄의 마지막으로 이동시키면 평균 종료시간을 더욱 감소시키기 때문에 작업 1 앞에 스케줄 된 하나의 작업을 뒤의 작업들과 교환할 경우만 고려하면 된다.)

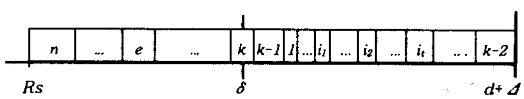

그림 4. 작업 1의 전후에 있는 작업을 교환

작업 e는 스케줄의 V-모양을 유지하기 위하여 마지막에 스케줄 되며 이때 종료시간의 합의 변화는 $-\Sigma_{j=1}^{e-1}(p_e - p_j)$ 이다. 작업들 $i_1, i_2, \ldots, i_l$ 는 작업 1의 앞에 스케줄 되며 이때 작업 i_k를 이동함으로써 생기는 종료시간의 변화는 $\Sigma_{j=1}^{i_k-1}(p_{i_k} - p_j)$ 이다. 그러므로, 작업들을 교환하므로써 생기는 작업 종료시간의 전체 변화량 ΔC는

$$
\begin{aligned}
\Delta C &= -\sum_{j=1}^{e-1}(p_e - p_j) + \sum_{k=1}^{l}\sum_{j=1}^{i_k-1}(p_{i_k} - p_j) \\
&= -\sum_{j=1}^{e-1}p_e + \sum_{j=1}^{e-1}p_j + \sum_{k=1}^{l}(i_k - 1)p_{i_k} - \sum_{k=1}^{l}\sum_{j=1}^{i_k-1}p_j \\
&= -(i_1 - 1)\Big(\sum_{k=1}^{l}p_{i_k} - p_e\Big) + \sum_{k=2}^{l}(i_k - i_1)p_{i_k} \\
&\quad -(e - i_1)p_e + \sum_{j=1}^{e-1}p_j - \sum_{k=1}^{l}\sum_{j=1}^{i_k-1}p_j \\
&\le (i_1 - 1)\Big(\sum_{k=1}^{l}p_{i_k} - p_e\Big) + (i_l - i_1)\Big(\sum_{k=2}^{l}p_{i_k} - p_e\Big) \\
&\quad -(e - i_l)p_e + \sum_{j=1}^{e-1}p_j - \sum_{k=1}^{l}\sum_{j=1}^{i_k-1}p_j
\end{aligned}
$$

작업 1을 왼쪽으로 이동시키기 위해서는 $(\Sigma_{k=1}^{l}p_{i_k} - p_e)$가 음수이어야 한다. 그러므로 ΔC는 음수가 되고 따라서 평균 종료시간도 감소한다. ♣

정리 4. *보조 정리 2에서 정의된 k 값을 사용하여 $\chi = \frac{1}{n}\{np_n + \ldots + kp_k + (k-1)p_{k-1} + (k-2)p_1 + \ldots + p_{k-2}\}$를 정의하자. 최대 허용 지연시간이 $MS - \chi$ 보다 작거나 같은 MSD/T_{max} 문제는 tightly Δ-constrained 이다.*

증 명. 다음에서 보이는 바와 같이 $\Delta \le MS - \chi$ 는 $d \ge \overline{C_u}$를 의미한다.

$$
\begin{aligned}
\Delta &\le MS - \chi \\
\Delta &\le MS - (\overline{C_u} - R_s) \\
\overline{C_u} &\le MS + R_s - \Delta \\
\overline{C_u} &\le d
\end{aligned}
$$

위의 정리를 증명하기 위해서, $\Delta \le MS - \chi$ 인 MSD/T_{max} 문제에서 최적 스케줄의 최대 지연시간이 최대 허용 지연시간보다 작다고 가정하자. 그러면, 이 스케줄이 가지는 작업들의 평균 종료시간은 보조 정리 2에 의해서 $\overline{C_u}$ 보다 작다. 따라서 이 스케줄을 오른쪽으로 이동하여 작업들의 평균 종료시간을 $\overline{C_u}$에 더 가까이 하면, [1]과 [3]에서 보여주듯이 MSD의 값이 더 작아진다. 따라서, 이 스케줄이 최적해라는 것은 모순이다.

5. 알고리즘

Δ_u와 Δ_l을 각각 다음과 같이 정의하자: $\Delta_u = MS - \chi$, $\Delta_l = (1/2)(MS - p_n)$. 그러면, 정리 3과 4에 의해 주어진 MSD/T_{max} 문제는 최대 허용 지연시간이 Δ_u보다 크면 Δ-unconstrained인 경우이고 Δ_l보다 작으면 tightly Δ-constrained인 경우라는 것을 알 수 있다. 따라서 Δ_u와 Δ_l은 각각 Δ-unconstrained와 Δ-constrained를 구별하는 上限과 Δ-constrained와 tightly Δ-constrained를 구별하는 下限이다. 우리는 이 값을 이용해 주어진 최대 허용 지연시간이 Δ인 MSD/T_{max} 의 최적 스케줄을 구하는 알고리즘을 기존의 알고리즘을 사용하여 쉽게 나타낼 수 있다. 다음의 알고리즘에서 CTV 알고리즘과 constrained MSD 알고리즘은 각각 [1]에서 제안된 unconstrained MSD 알고리즘과 constrained MSD 알고리즘을 나타내며, Δ-CCTV 알고리즘은 최대 허용 지연시간의 제약조건을 만족하면서 CTV를 가장 작게 하는 스케줄을 구하는 것으로 CTV 알고리즘을 변형하여 쉽게 얻을 수 있다.

단계 1. Δ_u 와 Δ_l 값을 계산한다.

단계 2. 만약 $\Delta \ge \Delta_u$이면 CTV 알고리즘으로 최적해를 구한다. 알고리즘을 마친다.

단계 3. 만약 $\Delta \le \Delta_l$ 이면 공통납기가 Δ이고 n-1개의 작업(가장 큰 작업을 제외한 작업들)을 가진 MSD 문제를 constrained MSD 알고리즘을 사용하여 최적해를 구한다. 여기서 구한 최적해를 역으로 하고 제외했던 가장 큰 작업을 앞쪽에 붙인 스케줄이 MSD/T_{max} 문제의 최적해이다.(단, 이 스케줄의 시작 시간은 R_s이다.) 알고리즘을 마친다.

단계 4. Δ-CCTV 알고리즘과 constrained MSD 알고리즘을 둘 다 수행하여 목적함수 값이 더 작은 스케줄이 MSD/T_{max} 문제의 최적해이다.

참고문헌

[1] Bagchi, U., R. S. Sullivan and Y. L. Chang, "Minimizing Mean Squared Deviation of Completion Times about a Common Due Date," Management Science 33, 894-906, 1987.

[2] Baker, K. R., G. D. Scudder, "Sequencing with Earliness and Tardiness Penalties: A Review," Operation Research 38, 22-36, 1990.

[3] De, P., J. B. Ghosh and C. E. Wells, "A note on the Minimization of Mean Squared Deviation of Completion Times About a Common Due Date," Management Science 35, 1143-1147, 1989.

[4] Kanet, J. J., "Minimizing the Average Deviation of Job Completion times about a Common Due Date," Naval Research Logistics Quarterly 28, 643-651, 1981.

[5] Merten, A. G. and M. E. Muller, "Variance Minimization in Single Machine Sequencing Problems," Management Science 18,518-528, 1972.

[6] Sidney, J. B., "Single-Machine Scheduling with Earliness and Tardiness Penalties," Operation Research 25, 62-69, 1977