한국경영과학회지
제38권 제2호
2013년 6월
http://dx.doi.org/10.7737/JKORMS.2013.38.2.025

불확실한 환경 하에서 중간 평가가 있는 시간-비용 프로젝트 문제*

최병천** · 정지복***†

Project Time-Cost Tradeoff Problem with Milestones under an Uncertain Processing Time

Byung-Cheon Choi** · Jibok Chung***

■ Abstract ■

We consider a project time-cost tradeoff problem with two milestones, where one of the jobs has an uncertain processing time. Unless each milestone is completed on time, some penalty cost may be imposed. However, the penalty costs can be avoided by compressing the processing times of some jobs, which requires additional resources or costs. The objective is to minimize the expected total costs subject to the constraint on the expected project completion time. We show that the problem can be solved in polynomial time if the precedence graph of a project is a chain.

Keyword : Time-Cost Tradeoff, Milestone, Uncertain Job

1. Introduction

In the time-cost tradeoff problem (TCTP), it is assumed that the processing times can be compressed through the expenditure of additional resources such as labor, capital and so on

논문접수일 : 2012년 10월 05일 논문게재확정일 : 2012년 12월 10일
논문수정일(1차 : 2012년 11월 14일, 2차 : 2012년 12월 05일)
* This study was financially supported by the research fund of Korea Sanhak Foundation in 2012.
This study was financially supported by the Daejeon University in 2011.
** Department of Business Administration, Chungnam National University
*** Department of Business Administration, Daejeon University
† Corresponding Author, jbchung@dju.ac.kr

[11, 17]. The typical objective of the TCTP is to minimize the project completion time subject to the constraint on the total costs or to minimize the total costs subject to the constraint on the project completion time.

Let the *linear* TCTP be defined as the TCTP such that the set of the possible processing times is a closed interval and the compression cost is decreasing linearly on the closed interval, and the *discrete* TCTP be defined as the TCTP such that the set of the possible processing times is discrete. The linear TCTP can be formulated as the linear programming (LP) problem and, furthermore, it is solvable by a network flow approach [6, 12]. However, the computational complexity of discrete TCTP is strongly NP-hard [5]. Thus, for the discrete TCTP, Weglarz et al. [19] discussed exact and approximate solution strategies and Skutella [18] developed the approximation algorithms with constant performance guarantee on various special cases.

The TCTP researches above assume that the processing times are deterministic. In reality, however, the processing time of a job can be affected by uncertain factors such as weather conditions, preventive maintenance and so on. This results in the stochastic TCTP (STCTP) such that the processing times are uncertain. Note that in this case, the mean value of the processing time can be compressed by additional resources. Let the *linear* STCTP be defined as the STCTP such that the set of the mean values of the possible processing times is a closed interval and the compression cost is linear and decreasing on the closed interval, and the *discrete* STCTP be defined as the STCTP such that the set of the mean values of the possible processing times is discrete.

Wollmer [20], Bowman [3] and Leu et al. [13] considered the linear STCTP in which the objective is to minimize the expected project completion time subject to the constraint on the total compression costs or to minimize the total compression costs subject to the constraint on the expected project completion time.

Wollmer [20] formulated the problem as a stochastic programming, and developed the cutting plan technique which converges to an optimal schedule. Bowman [3] presented a heuristic algorithm based on simulation technique by using the derivative estimators. Leu et al. [13] described the uncertainties of the processing times through Fuzzy set theory, and presented the searching algorithm based on genetic algorithms. Cohen et al. [4] considered the linear STCTP in which the objective is to minimize the expected total costs under a specified due-date of a project. The expected costs are described as the sum of the total compression cost and a time-related overhead cost. They introduced a new model based on the robust optimization, which was developed by Ben-Tal and Nemirovski [2] for solving large-scale convex optimization problems in which portions of the data are uncertain and known only to vary within given uncertainty sets, and presented its potential benefits. Mokhtari et al. [14] considered the linear STCTP in which the objective is to minimize the total compression costs subject to the constraint on the probability that the project is terminated before a predetermined deadline. They developed a hybrid approach based on the cutting plane method and Monte Carlo simulation. Gutjahr et al. [8] considered the discrete STCTP in which the objective is to minimize the expected total costs, which are expressed as the expected penalty cost for the project completion plus the total compression costs. They presented a stochastic branch-and-bound procedure. Hazir et al. [10] considered the discrete STCTP to maximize the expected profit under a given budget. They introduced surrogate measures of providing an accurate estimate of the scheduling robustness.

The TCTP researches above assume that throughout the overall project, there exists only one milestone, that is, the time when the project has been terminated. In reality, however, there may exist some milestones in the middle of the project. For example, a venture capital company makes small investments at first, and then determines whether to stop the project or make more investment, when a milestone has been reached [1, 16]. To our best knowledge, however, no paper has been conducted on the STCTP with more than two milestones.

In this paper, we consider the linear STCTP with two milestones such that some penalty costs occur if each milestone is not reached at the appointed due date. The objective is to minimize the expected total costs subject to the constraint on the expected project completion time, where the expected total costs are expressed as the expected total penalty costs plus and the total compression costs.

In the linear STCTP with a general precedence graph, however, computing the expected project completion time is #P-complete, even when the mean value of each processing time cannot be compressed [9]. Note that any #P-complete problem is polynomially equivalent to determining the number of Hamiltonian circuits in a graph [7]. This implies that since the constraint of our problem is to check whether the expected project completion time is larger than a specific threshold, our problem is at least #P-complete. Thus, we consider the special but practical case such that

- The precedence graph of a project is a chain. This property is motivated from a product development process that consists of sequential stages [15];
- One job has an uncertain processing time which is distributed according to the uniform distribution. This property is motivated from a project where a new technology is applied to one of jobs and the existing technologies are applied to the others. Since probabilistic information for the new technology is not sufficiently accumulated yet, furthermore, the uncertainty of the processing time which is implemented by the new technology can be described by an interval [4].

The remainder of this paper is organized as follows. Section 2 presents the problem definition in formal terms. In Section 3, we present an $O(n^2)$ algorithm for the LP problem reducible to our problem. Section 4 shows that our problem is solvable in $O(n^2)$ by using the results of Section 3. Finally, we complete the paper with concluding remarks and future works.

2. Problem Definition

Our problem can be formally stated as follows. A precedence graph $G=(N, A)$ of a project is a chain, described below, in which $N=1,\ 2,\ \cdots,\ n$ is the set of jobs, and A is the set of precedence relations between jobs :

$$1 \rightarrow 2 \rightarrow \cdots \rightarrow n.$$

Note that $i \rightarrow j$ means job j can be started after job i has been completed. Let jobs m and n be two milestones, where $m<n$. For $j=m,\ n$, let w_j and d_j be the penalty cost and the appointed time of job j, respectively. Note that if job j is finished after d_j, then penalty cost w_j occurs for $j=m,\ n$. Let $M=1,\ 2,\cdots,\ m$. Let job k be an index for an uncertain job. The processing time of job k is a random variable P_k that is distributed according to a uniform distribution $f(p_k)$ with two parameters a_k and b_k, where a_k and b_k are minimum and maximum values, respectively, that is,

$$f(p_k)=\frac{1}{b_k-a_k},\quad a_k\le p_k\le b_k.$$

The initial processing time of deterministic job j is p_j for $j\in N\setminus\{k\}$. Associated with job j is a maximal amount for compression u_j and a compression cost rate c_j, $j=1,\ 2,\ \cdots,\ n$. It is assume that $a_k-u_k\ge 0$. Let $x=(x_1,\ x_2,\ \cdots,\ x_n)$ be the schedule, where x_j is the compressed amount of job j and $0\le x_j\le \mathrm{u_j}$, $j=1,\ 2,\ \cdots,\ n$. Let $g_j(x_j)=c_jx_j$ be the compression cost of job j, $j=1,2,\cdots,n$. Note that if job k is compressed by x_k, then p_k is distributed according to a uniform distribution $f(\bar{p}_k)$, defined below :

$$f(\bar{p}_k)=\frac{1}{b_k-a_k},\quad a_k-x_k\le\bar{p}_k\le b_k-x_k.$$

The completion time of job j in x is a random variable $\mathsf{C}_m(c)$, $j=1,\ 2,\ \cdots,\ n$. Then, our problem is defined below :

$$\begin{aligned}\text{minimize}\quad & w_mP(\mathsf{C}(x)>d_m)+w_nP(\mathsf{C}_n(x)>d_n)\\ & +\textstyle\sum_{j=1}^{n}c_jx_j\\ \text{subject to}\quad & E(\mathsf{C}_n(x))\le d\\ & 0\le x_j\le u_j,\ j=1,\ 2,\cdots,\ n,\end{aligned}$$

where $P(\mathsf{C}_j(x)>d_j)$ is the probability that the job j is completed after d_j, $j=m,n$, $E(\mathsf{C}_n(x))$ is the expected value of the project completion time and d is the deadline of the project completion time. Let our problem be referred to as **Problem P.**

3. LP Problem with Special Constraints

In this section, we introduce the LP problem with the special structure of constraints, and construct an algorithm for the LP problem, defined below :

$$\begin{aligned}\text{minimize}\quad & z(x)=\textstyle\sum_{j=1}^{n}c_jx_j\\ \text{subject to}\quad & a_1\le\textstyle\sum_{j=1}^{n}x_j\le b_1\\ & a_2\le\textstyle\sum_{j=1}^{n}x_j\le b_2\\ & 0\le x_j\le u_j,\ j=1,\ 2,\cdots,\ n,\end{aligned}\tag{1}$$

where $m<n$. Since $\sum_{j=1}^{m}x_j<\sum_{j=1}^{n}x_j$, without loss of generality, we can assume that $a_1\le a_2$ and $b_1\le b_2$. Let LP problem (1) be referred to as LPSC. Let $c_g=\min\{c_j|\ j=1,\ 2,\cdots,\ m\}$. Henceforth, we will show that the LPSC with $a_1\ne 0$ can be reduced to the LPSC with $a_1=0$.

Lemma 1 : *If* $a_1>0$, *then there exists an optimal schedule,* x^* *in the LPSC such that* $x_g^*\ge\min\ \{u_g,\ a_1\}$.

Proof. There exists an optimal schedule, x^* such that $x_g^*<\min\{u_g,\ a_1\}$. Since $\sum_{j=1}^{m}x_j^*\ge a_1$, there exists a vector, $\epsilon^a=(\epsilon_1^a,\ \epsilon_2^a,\ \cdots,\ \epsilon_n^a)$ such that

- $\epsilon_g^a=\min\{u_g,\ a_1\}-x_g^*$;
- $\sum_{j\in M\setminus\{g\}}\epsilon_j^a=\epsilon_g^a$ and $\epsilon_j^a\ge 0$ for $j\in M\setminus\{g\}$;
- $x_j^*-\epsilon_j^a\ge 0$ for $j\in M\setminus\{g\}$.

Then, we can construct a new schedule, x^a

such that

$$x_j^a = \begin{cases} x_j^* - \epsilon_j^a & \text{for } j \in M \setminus \{g\} \\ x_j^* + \epsilon_j^a & \text{for } j = g \\ x_j^* & \text{for } j = m+1,\ m+2,\ \cdots,\ n. \end{cases}$$

Since $x_a^g \le u_g$, $\sum_{j=1}^m x_j^* = \sum_{j=1}^m x_j^a$ and $\sum_{j=1}^n x_j^* = \sum_{j=1}^n x_j^a$, x^a is a feasible schedule. Furthermore, since $c_g \le \min\{c_j | \ j=1,\ 2, \cdots,\ m\}$ and $\epsilon_g^a = \sum_{j \in M \setminus g} \epsilon_j^a$,

$$z(x^*) - z(x^a) = -c_g \epsilon_g^a + \sum_{j \in M \setminus \{g\}} c_j \epsilon_j^a \ge 0.$$

Thus, we can construct another optimal schedule satisfying Lemma 1 from x^* without increasing the objective value. □

Let $\sigma_1 = (\sigma_1(1),\ \sigma_1(2),\ \cdots,\ \sigma_1(m))$ be the permutation such that

- $\{\sigma_1(1),\ \sigma_1(2),\ \cdots,\ \sigma_1(m)\} = \{1,\ 2,\ \cdots,\ m\}$;
- $c_{\sigma_1(1)} \le c_{\sigma_1(2)} \le \cdots \le c_{\sigma_1(m)}$;
- If $c_{j_1} = c_{j_2}$ and $j_1 < j_2$, then $\sigma_1^{-1}(j_1) < \sigma_1^{-1}(j_2)$, where $\sigma_1^{-1}(j)$ is the order of job j in σ_1.

Let h_1 be the index such that $\sum_{j=0}^{h_1} u_{\sigma_1(j)} \le a_1 < \sum_{j=0}^{h_1+1} u_{\sigma_1(j)}$, where, for consistency of notations, let $u_{\sigma_1(0)} = 0$. By Lemma 1 and $x_j \le u_j$, $j = 1,\ 2,\ \cdots,\ n$, it is observed that there exists an optimal schedule, x^* such that

$$x^*_{\sigma_1(j)} = u_{\sigma_1(j)}, \quad j = 1,\ 2,\ \cdots,\ h_1 \text{ and}$$
$$x^*_{\sigma_1(h_1+1)} \ge a_1 - \sum_{j=1}^{h_1} u_{\sigma_1(j)}.$$

Let $M' = \{\sigma_1(1),\ \sigma_1(2),\ \cdots,\ \sigma_1(\mathrm{h}_1)\}$. Then, based on the above observations, the LPSC can be written as follows :

$$\text{minimize } \sum_{j \in N \setminus M'} c_j x_j$$
$$\text{subject to } 0 \le \sum_{j \in M \setminus M'} x_j \le b_1 - a_1$$
$$a_2 - a_1 \le \sum_{j \in N \setminus M'} x_j \le b_2 - a_1$$
$$0 \le x_j \le u_j \text{ for } j \in N \setminus (M' \cup \{\sigma_1(h_1+1)\})$$
$$0 \le x_j \le u_j - (a_1 - \sum_{j=1}^{h_1} u_{\sigma_1(j)})$$
$$\text{for } j = \sigma_1(h_1+1).$$

It is observed that LP problem (1) can be reduced to LP with $a_1 = 0$. Henceforth, without loss of generality, assume that $a_1 = 0$ in the LPSC.

3.1 Optimality Conditions for the LPSC

In this subsection, we introduce two optimality conditions, which will be used to construct an algorithm for the LPSC. Let

$$c_l = \begin{cases} \min\{c_j | \ j = 1,\ 2, \cdots,\ n\} & \text{if } b_1 > 0 \\ \min\{c_j | \ j = m+1,\ m+2,\ \cdots,\ n\} & \text{if } b_1 = 0. \end{cases}$$

First, we present the property of an optimal schedule (Lemma 1) for the case that the sign of c_l is negative, and then the one of an optimal schedule for the case that the sign of c_l is non-negative (Lemma 2).

Lemma 2 : *If $c_l < 0$, then there exists an optimal schedule, x^* in the LPSC such that*

i) If $1 \le l \le m$, then $x_l^* = \min\{u_l,\ b_1\}$;
ii) If $m+1 \le l \le n$, then $x_l^* = \min\{u_l,\ b_2\}$.

Proof. The proof is given in Appendix A. □

Lemma 3 : *If $c_l \ge 0$, then there exists an optimal* *schedule, x^* in the LPSC such that*

i) If $1 \le l \le m$, then $x_l^* = \min\{u_l,\ b_1,\ a_2\}$;

ii) If $m+1 \le l \le n$, then $x_l^* = \min\{u_l,\ a_2\}$.

Proof. The proof is given in Appendix B. □

The framework of two proofs above is to construct a new optimal schedule satisfying Lemma 2 or Lemma 3, or verify the contradiction under the assumption that there exists an optimal schedule which does not satisfy Lemma 2 or Lemma 3.

3.2 Algorithm for the LPSC

In this section, we develop the algorithm for the LPSC. Recall that $a_1 = 0$. Then, based on Lemmas 2 and 3, we can construct the algorithm below:

Algorithm OPT

Step 0 Set $N_1 = \{1,\ 2,\ \cdots,\ m\}$, $N_2 = \{m+1,\ m+2, \cdots,\ n\}$, $x_j = 0$, $j = 1,\ 2,\ \cdots,\ n$, $\alpha_2 = a_2$ and $\beta_i = b_i$, $i = 1,\ 2$.

Step 1 If $\beta_1 > 0$, then find job l such that $c_l = \min\ \{c_j |\ j \in N_1 \cup N_2\}$, while if $\beta_1 = 0$, find job l such that $c_l = \min\{c_j |\ j \in N_2\}$.

Step 2 If $c_l \ge 0$, then go to Step 2-1, while if $c_l < 0$, then go to Step 2-2.

Step 2-1 If $l \in N_1$, then set $x_l = \min\{u_l,\ \beta_1,\ \alpha_2\}$, while if $l \in N_2$, then set $x_l = \min\{u_l,\ \alpha_2\}$. If x is feasible in the LPSC, then x is an optimal schedule and STOP.

Step 2-2 If $l \in N_1$, then let $x_l = \min\{u_l,\ \beta_1\}$, while if $l \in N_2$, then let $x_l = \min\{u_l,\ \beta_2\}$.

Step 3 If $l \in N_1$, then go to Step 3-1, while if $l \in N_2$, then go to Step 3-2 :

Step 3-1 Set $N_1 = N_1 \setminus \{l\} \beta_1 = \beta_1 - x_l$, $\alpha_2 = \max\{0,\ \alpha_2 - x_l\}$ and $\beta_2 = \beta_2 - x_l$.

Step 3-2 Set $N_2 = N_2 \setminus \{l\}$, $\alpha_2 = \{\max\ 0,\ \alpha_2 - x_l\}$ and $\beta_2 = \beta_2 - x_l$. If $\beta_1 > \beta_2$, then set $\beta_1 = \beta_2$.

Step 4 If $N_1 = \varnothing$ and $N_2 = \varnothing$, then go to Step 6, while otherwise, go to Step 5.

Step 5 If $\beta_1 = 0$ and $\beta_2 = 0$, go to Step 6, while otherwise, go to Step 1.

Step 6 If x is feasible in the LPSC, then x is an optimal schedule and STOP. Otherwise, the LPSC is infeasible and STOP.

Note that the time complexity of Algorithm OPT runs in $O(n^2)$.

Theorem 1 : *The LPSC can be solved in $O(n^2)$.*

3.3 Numerical example

We illustrate the use of Algorithm OPT by means of a numerical example below :

$$\begin{aligned} &\text{minimize} && -2x_1 + 4x_2 - x_3 + 2x_4 + 4x_5 \\ &\text{subject to} && 0 \le x_1 + x_2 + x_3 \le 3 \\ &&& 2 \le x_1 + x_2 + x_3 + x_4 + x_5 \le 5 \\ &&& 0 \le x_1 \le 2,\ 0 \le x_2 \le 2,\ 0 \le x_3 \le 2, \\ &&& 0 \le x_4 \le 1,\ 0 \le x_5 \le 3. \end{aligned}$$

This problem can be solved as follows :

Step 0 $N_1 = \{1,\ 2,\ 3\}$ and $N_2 = \{4,\ 5\}$, $\alpha_2 = 2$, $\beta_1 = 3$. and $\beta_2 = 5$.

Stage 1 :

Step 1 $l = 1$ and $c_l = -2$.

Step 2 Since $c_l < 0$, go to Step 2-2.

Step 2-2 Since $l \in N_1$, $x_1 = \min\{2,\ 3\} = 2$.

Step 3 Since $l \in N_1$, go to Step 3-1.

Step 3-1 $N_1 = \{2,\ 3\}$, $\beta_1 = 1$, $\alpha_2 = 0$ and $\beta_2 = 3$.

Stage 2 :

Step 1 $l = 3$ and $c_l = -1$.

Step 2 Since $c_l < 0$, go to Step 2-2.

Step 2-2 Since $l \in N_1$, $x_3 = \min\{2,\ 1\} = 1$.

Step 3 Since $l \in N_1$, go to Step 3-1.

Step 3-1 $N_1 = \{2\}$, $\beta_1 = 0$, $\alpha_2 = 0$ and $\beta_2 = 2$.

Stage 3 :

Step 1 $l = 4$ and $c_l = 2$.

Step 2 Since $c_l \geq 0$, go to Step 2-1.

Step 2-1 Since $l \in N_2$, $x_4 = \min\{1,\ 0\} = 0$.
Since $x = (2,\ 0,\ 1,\ 0,\ 0,\ 0)$ is feasible, x is an optimal and STOP.

4. Polynomiality of Problem P

In this section, we show that Problem P is polynomially solvable in $O(n^2)$ through the decomposition of Problem P into several LP problems which belong to the class of the LPSC. Since the expression $P(\mathrm{C}_m(x) \leq d_m)$ is different depending on the position of job k, we consider two cases below.

Case 1: Job m is positioned after job k, that is, $m \geq k$

In this case, $\mathrm{C}_m(x)$ is a random variable. Thus, $P(\mathrm{C}_m(x) \leq d_m)$ is calculated as follows : Let $\overline{M} = \{1,\ 2,\ \cdots,\ m\} \setminus \{k\}$.

i) $d_m < a_k - x_k + \sum_{j \in \overline{M}} (p_j - x_j)$

$$P(\mathrm{C}_m(x) \leq d_m) = 0$$

ii) $a_k - x_k + \sum_{j \in \overline{M}} (p_j - x_j) \leq$
$d_m \leq b_k - x_k + \sum_{j \in \overline{M}} (p_j - x_j)$

$$P(\mathrm{C}_m(x) \leq d_m) = \int_{a_k - x_k + \sum_{j \in \overline{M}} (p_j - x_j)}^{d_m} \frac{1}{b_k - a_k} d\,\mathrm{C}_m(x) = \frac{\sum_{j=1}^{m} x_j - a_k - \sum_{j \in \overline{M}} p_j + d_m}{b_k - a_k}.$$

iii) $b_k - x_k + \sum_{j \in \overline{M}} (p_j - x_j) < d_m$

$$P(\mathrm{C}_m(x) \leq d_m) = 1$$

For simplicity of notations, let $F_m(\sum_{j=1}^{m} x_j) = P(\mathrm{C}_m(x) \leq d_m)$, $q_m^1 = a_k + \sum_{j \in \overline{M}} p_j - d_m$ and $q_m^2 = b_k + \sum_{j \in \overline{M}} p_j - d_m$. Then,

$$F_m(\sum_{j=1}^{m} x_j) = \begin{cases} 0 & \text{if } \sum_{j=1}^{m} x_j < q_m^1 \\ \dfrac{\sum_{j=1}^{m} x_j - q_m^1}{q_m^2 - q_m^1} & \text{if } q_m^1 \leq \sum_{j=1}^{m} x_j \leq q_m^2 \\ 1 & \text{if } q_m^2 < \sum_{j=1}^{m} x_j \end{cases}$$

Case 2 : Job m is positioned before job k, that is, $m < k$

In this case, $\mathrm{C}_m(x)$ is a deterministic variable. Thus, $F_m(\sum_{j=1}^{m} x_j)$ is calculated as follows : Let $q_m = \sum_{j=1}^{m} p_j - d_m$.

$$F_m(\sum_{j=1}^{m} x_j) = \begin{cases} 0 & \text{if } \sum_{j=1}^{m} x_j < q_m \\ 1 & \text{if } \sum_{j=1}^{m} x_j \geq q_m. \end{cases}$$

Henceforth, we will describe the expressions of $P(\mathrm{C}_n(x) \leq d_n)$ and $E(\mathrm{C}_n(x))$. Note that since $n \geq k$, $\mathrm{C}_n(x)$ is a random variable. Thus, by using the way applied to Case 1, $P(\mathrm{C}_n(x) \leq d_n)$ is calculated as follows : let $F_n(\sum_{j=1}^{n} x_j) = P(\mathrm{C}_n(x) \leq d_n)$, $q_n^1 = a_k + \sum_{j \in \overline{N}} p_j - d_n$ and $q_n^2 = b_k + \sum_{j \in \overline{N}} p_j - d_n$, where $\overline{N} = \{1,\ 2,\ \cdots,\ n\} \setminus \{k\}$.

$$F_n(\sum_{j=1}^{n} x_j) = \begin{cases} 0 & \text{if } \sum_{j=1}^{n} x_j < q_n^1 \\ \dfrac{\sum_{j=1}^{n} x_j - q_n^1}{q_n^2 - q_n^1} & \text{if } q_n^1 \leq \sum_{j=1}^{n} x_j \leq q_n^2 \\ 1 & \text{if } q_n^2 < \sum_{j=1}^{n} x_j. \end{cases}$$

$E(\mathrm{C}_n(x))$ is calculated as follows :

$$E(\mathrm{C}_n(x)) = \int_{\sum_{j\in N\setminus\{k\}}p_j + a_k - \sum_{j=1}^{n}x_j}^{\sum_{j\in N\setminus\{k\}}p_j + b_k - \sum_{j=1}^{n}x_j} \frac{\mathrm{C_n(x)}}{b_k - a_k} d\mathrm{C_n(x)}$$

$$= \sum_{j\in N\setminus\{k\}} p_j + \frac{1}{2}(a_k + b_k) - \sum_{j=1}^{n} x_j.$$

Thus, we will reformulate Problem P by using the expressions of $P(\mathrm{C}_m(x) \le d_m)$, $P(\mathrm{C}_n(x) \le d_m)$ and $E(\mathrm{C}_n(x))$. Since $F_j(\sum_{j=1}^{n}x_j) = 1 - P(\mathrm{C}_n(x) > d_n)$ and w_j is a constant for $j \in \{m, n\}$, Problem P can be rewritten below : Let $\bar{d} = \sum_{j\in N\setminus\{k\}} p_j + \frac{1}{2}(a_k + b_k) - d.$

$$\begin{aligned} \text{minimize } & -w_m F_m(\textstyle\sum_{j=1}^{m}x_j) - w_n F_n(\sum_{j=1}^{n}x_j) \\ & + \textstyle\sum_{j=1}^{n} c_j x_j \\ \text{subject to } & x \in Q_d, \end{aligned}$$

$$\begin{aligned} \text{where } Q_d = & \{x \mid \bar{d} \le \textstyle\sum_{j=1}^{n}x_j\} \cap \\ & \{x \mid 0 \le x_j \le u_j,\ j = 1, 2, \cdots, n\}. \end{aligned}$$

Theorem 2 : *When job m is processed after job k $(m > k)$, then Problem P can be solved in $O(n^2)$.*

Proof. In this case, Problem P can be written as follows:

$$\begin{aligned} \text{minimize } & -w_m F_m(\textstyle\sum_{j=1}^{m}x_j) - w_n F_n(\sum_{j=1}^{n}x_j) \\ & + \textstyle\sum_{j=1}^{n} c_j x_j \\ \text{subject to } & x \in Q_d, \end{aligned}$$

where

$$F_m(\sum_{j=1}^{m}x_j) = \begin{cases} 0 & \text{if } \sum_{j=1}^{m}x_j < q_m^1 \\ \dfrac{\sum_{j=1}^{m}x_j - q_m^1}{q_m^2 - q_m^1} & \text{if } q_m^1 \le \sum_{j=1}^{m}x_j \le q_m^2 \\ 1 & \text{if } q_m^2 < \sum_{j=1}^{m}x_j \end{cases}$$

and

$$F_n(\sum_{j=1}^{n}x_j) = \begin{cases} 0 & \text{if } \sum_{j=1}^{n}x_j < q_n^1 \\ \dfrac{\sum_{j=1}^{n}x_j - q_n^1}{q_n^2 - q_n^1} & \text{if } q_n^1 \le \sum_{j=1}^{n}x_j \le q_n^2 \\ 1 & \text{if } q_n^2 < \sum_{j=1}^{n}x_j. \end{cases}$$

Consider six sets of schedules below :

$$Q_m^1 = \{x \mid 0 \le \sum_{j=1}^{m}x_j \le q_m^1\},$$

$$Q_m^2 = \{x \mid q_m^1 \le \sum_{j=1}^{m}x_j \le q_m^2\}$$

$$\text{and } Q_m^3 = \{x \mid q_m^2 \le \sum_{j=1}^{m}x_j \le Q_m\},$$

and

$$Q_n^1 = \{x \mid 0 \le \sum_{j=1}^{n}x_j \le q_n^1\},$$

$$Q_n^2 = \{x \mid q_n^1 \le \sum_{j=1}^{n}x_j \le q_n^2\}$$

$$\text{and } Q_n^3 = \{x \mid q_n^2 \le \sum_{j=1}^{n}x_j \le Q_n\},$$

where let $Q_m = \sum_{j=1}^{m}u_j$ and $Q_n = \sum_{j=1}^{n}u_n$. It is observed that there exists an optimal schedule in one of nine feasible regions below :

$$Q_d \cap Q_m^{i_m} \cap Q_n^{i_n} \quad \text{for } i_m \in \{1, 2, 3\} \text{ and } i_n \in \{1, 2, 3\}.$$

Furthermore, since the objective function corresponding to each feasible region is linear, nine LP's have the forms of the LPSC. Thus, since nine LP's can be solved by Algorithm OPT, the proof is complete. □

Theorem 3 : *When job m is processed before job k, then Problem P can be solved in $O(n^2)$.*

Proof. In this case, the problem can be written as follows :

$$\text{minimize } -w_m F_m(\textstyle\sum_{j=1}^{m}x_j) - w_n F_n(\sum_{j=1}^{n}x_j)$$

$+\sum_{j=1}^{n} c_j x_j$

subject to $x \in Q_d$,

where

$$F_m(\sum_{j=1}^{m} x_j)=\begin{cases}0 & \text{if } \sum_{j=1}^{m} x_j < q_m \\ 1 & \text{if } \sum_{j=1}^{m} x_j \geq q_m\end{cases}$$

and

$$F_n(\sum_{j=1}^{n} x_j)=\begin{cases}0 & \text{if } \sum_{j=1}^{n} x_j < q_n^1 \\ \dfrac{\sum_{j=1}^{n} x_j - q_n^1}{q_n^2 - q_n^1} & \text{if } q_n^1 \leq \sum_{j=1}^{n} x_j \leq q_n^2 \\ 1 & \text{if } q_n^2 < \sum_{j=1}^{n} x_j.\end{cases}$$

Consider five sets of schedules below :

$$Q_m^1 = \{x \mid 0 \leq \sum_{j=1}^{m} x_j \leq q_m^1\},$$

$$\text{and } Q_m^2 = \{x \mid q_m \leq \sum_{j=1}^{m} x_j \leq Q_m\}$$

and

$$Q_n^1 = \{x \mid 0 \leq \sum_{j=1}^{n} x_j \leq q_n^1\},$$

$$Q_n^2 = \{x \mid q_n^1 \leq \sum_{j=1}^{n} x_j \leq q_n^2\}$$

$$\text{and } Q_n^3 = \{x \mid q_n^2 \leq \sum_{j=1}^{n} x_j \leq Q_n\},$$

where let $Q_m = \sum_{j=1}^{m} u_j$ and $Q_n = \sum_{j=1}^{n} u_n$. It is observed that there exists an optimal schedule in one of six feasible regions below :

$$Q_d \cap Q_m^{i_m} \cap Q_n^{i_n} \quad \text{for } i_m \in \{1,\ 2\} \text{ and } i_n \in \{1,\ 2,\ 3\}.$$

Furthermore, since the objective function corresponding to each feasible region is linear, six LP's have the forms of LP problem (1). Thus, since six LP's can be solved by Algorithm OPT, the proof is complete. □

4.1 Numerical example

We illustrate the reduction of Theorem 2 by means of a numerical example below : Consider Problem P with five jobs such that the deadline is 20, job 4 is an uncertain job and the information of each job is as follows. $f(p_4)=\frac{1}{6}$ for $2 \leq p_4 \leq 8,\ m=4$ and

j	1	2	3	4	5
p_j	5	5	5	p_4	5
c_j	1	2	3	3	3
u_j	4	4	4	4	4
d_j	-	-	-	12	14
w_j	-	-	-	10	15

Then, Problem P can be reduced to the following problem.

minimize $-10F_4(\sum 1_{j=1}^{4} x_j) - 15F_5(\sum_{j=1}^{5} x_j)$
$+\sum_{j=1}^{5} c_j x_j$

subject to $x \in Q_d$,

where $Q_d = \{x \mid 5 \leq \sum_{j=1}^{5} x_j\} \cap$
$\{x \mid 0 \leq x_j \leq u_j,\ j=1,\ 2, \cdots,\ 5\}$,

$$F_4(\sum_{j=1}^{4} x_j)=\begin{cases}0 & \text{if } \sum_{j=1}^{4} x_j < 5 \\ \frac{1}{6}(\sum_{j=1}^{4} x_j - 5) & \text{if } 5 \leq \sum_{j=1}^{4} x_j \leq 11 \\ 1 & \text{if } 11 < \sum_{j=1}^{4} x_j.\end{cases}$$

and

$$F_5(\sum_{j=1}^{5} x_j)=\begin{cases}0 & \text{if } \sum_{j=1}^{5} x_j < 8 \\ \frac{1}{6}(\sum_{j=1}^{5} x_j - 8) & \text{if } 8 \leq \sum_{j=1}^{5} x_j \leq 14 \\ 1 & \text{if } 14 < \sum_{j=1}^{5} x_j\end{cases}$$

Let

$$Q_4^1 = \{x \mid 0 \le \sum_{j=1}^{4} x_j \le 5\},\ Q_4^2 = \{x \mid 5 \le \sum_{j=1}^{4} x_j \le 11\}$$

$$\text{and } Q_4^3 = \{x \mid 11 \le \sum_{j=1}^{4} x_j \le 16\},$$

and

$$Q_5^1 = \{x \mid 0 \le \sum_{j=1}^{5} x_j \le 8\},\ Q_5^2 = \{x \mid 8 \le \sum_{j=1}^{5} x_j \le 14\}$$

$$\text{and } Q_5^3 = \{x \mid 14 \le \sum_{j=1}^{5} x_j \le 20\}.$$

Then, it is observed that there exists an optimal schedule in one of nine feasible regions below :

$$Q_d \cap Q_4^{i_4} \cap Q_5^{i_5} \text{ for } i_4 \in \{1,\ 2,\ 3\} \text{ and } i_5 \in \{1,\ 2,\ 3\}.$$

5. Conclusion and Future Works

We consider the linear STCTP with two milestones such that the precedence graph is a chain. The objective is to minimize the expected total costs subject to the constraint on the expected project completion time. We show that our problem is solvable in $O(n^2)$ by reducing it to the special form of the LP problem, which can be solved in $O(n^2)$.

For future works, it would be interesting to extend the results of our research to a project such that the precedence graph is series-parallel or parallel, and the number of uncertain jobs is more than one.

References

[1] Bell, J., "Beauty is in the eye of the beholder : Establishing a fair and equitable value for embryonic high-tech enterprises," *Venture Capital Journal*, Vol.40(2000), pp.33-34.

[2] Ben-Tal, A. and A. Nemirovski, "A robust optimizatioin-Methodelogy and applications," *Mathematical Programming*, Vol.92(2002), pp.86-100.

[3] Bowman, R.A., "Stochastic gradient-based time-cost tradeoffs in pert networks using simulation," *Annals of Operations Research*, Vol.53(1994), pp.533-551.

[4] Cohen, I., B. Golany, and A. Shtub, "The stochastic time-cost tradeoff problem : A robust optimization approach," *Networks*, Vol.49 (2007) pp.175-188.

[5] De, P., J. Dunne, J.B. Ghosh, and C.E. Wells, "Complexity of the discrete time-cost tradeoff problem for project networks," *Operations Research*, Vol.45(1997), pp.302-306.

[6] Fulkerson, D.R. "A network flow computation for project cost curves," *Management Science*, Vol.7(1961), pp.167-178.

[7] Garey, M.R. and D.S. Johnson, *Computers and intractability : A guide to the theory of NP-completeness* W.H. Freeman, 1979.

[8] Gutjahr, W.J., C. Strauss, and E. Wagner, "A stochastic branch-and-bound approach to activity crashing in project management," *Informs Journal on Computing*, Vol.12(2000), pp.125-135.

[9] Hagstrom, J.N., "Computational complexity of pert problems," *Networks*, Vol.1(1988), pp. 139-147.

[10] Hazir, O., M. Haouari, and E. Erel, "Robust scheduling and robustness measures for the discrete time-cost trade-off problem," *European Journal of Operational Research*, Vol. 207(2010), pp.633-643.

[11] Hillier, F.S. and G.J. Lieberman, *Introduction to Operations Research*, newblock McGraw-

Hill, New York, NY, 2001.

[12] Kelley, J.E., "Critical path planning and scheduling : Mathematical basis," *Operations Research*, Vol.9(1961), pp.296-320.

[13] Leu, S.S., A.T. Chen, and C.H. Yang, "A GA-based fuzzy optimal model for construction time-cost trade-off," *International Journal of Project Management*, Vol.19(2001), pp. 47-58.

[14] Mokhtari, H., A. Aghaie, J. Rahimi, and A. Mozdgir, "Project time-cost trade-off scheduling : a hybid optimization approach," *International Journal of Advanced Manufacturing Technology*, Vol.50(2010), pp.811-822.

[15] Roemer, T. and R. Ahmadi, "Concurrent crashing and overlapping in product development," *Operations Research*, Vol.52(2004), pp.606-622.

[16] Sahlmann, W.A., "Insights from the ventures capital model of project governance," *Business Economics*, Vol.29(1994), pp.35-41.

[17] Shtub, A., J.F. Bard, and S. Globerson, *Project Managemnet : Engineering, Technology and Implementation*, Prince Hall, Englewood Cliff, NJ. (1994).

[18] Skutella, M., "Approximation algorithms for the discrete time-cost tradeoff problem," *Proceedings of the Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, New Orleans, L.A. (1997), pp.501-508.

[19] Weglarz, J., J. Jozefowska, M. Mika, and G. Waligora, "Project scheduling with finite or infinite number of activity processing modes -a survey," *European Journal of Operational Research*, Vol.208(2011), pp.177-205.

[20] Wollmer, R.D., "Critical path planning under uncertainty," *Mathematical Programming Study*, Vol.25(1985), pp.164-171.

<Appendix A>

Proof. i) Suppose that there exists an optimal schedule, x^* such that $x_l^* \neq \min\{u_l,\ b_1\}$. If $\sum_{j=1}^{n} x_j^* < \min\{u_l,\ b_1\}$, then we can construct a new schedule, x^a such that

$$x_j^a = \begin{cases} x_j^* & \text{for} \quad j \in N\{l\} \\ x_j^* + \min\{u_{l,\ b_1}\} - \sum_{j=1}^{n} x_j^* & \text{for} \quad j = l. \end{cases}$$

Since $x_l^a \leq u_l$, $\sum_{j=1}^{m} x_j^a \leq b_1$ and $\sum_{j=1}^{n} x_j^* \leq \sum_{j=1}^{n} x_j^a \leq b_2$, x^a is a feasible schedule. Furthermore, since $c_l < 0$ and min $\{u_{l,\ b_1}\} - \sum_{j=1}^{n} x_j^* > 0$,

$$z(x^*) - z(x^a) = -c_l(\min\{u_l,\ b_1\} - \sum_{j=1}^{n} x_j^*) > 0.$$

This is a contradiction. Thus, we observe that in an optimal schedule, x^*,

$$\sum_{j=1}^{n} x_j^* \geq \min\{u_l,\ b_1\}. \tag{2}$$

If $\sum_{j=1}^{m} x_j^* < \min\{u_l,\ b_1\}$, then, by inequality (2), there exists a vector, $\epsilon^b = (\epsilon_1^b,\ \epsilon_2^b,\ \cdots,\ \epsilon_n^b)$ such that

- $\epsilon_l^b = \min\{u_{l,}\ b_1\} - \sum_{j=1}^{m} x_j^*$;
- $\sum_{j=m+1}^{n} \epsilon_j^b = \epsilon_l^b$ and $\epsilon_j^b \geq 0,\ j = m+1, m+2,\ \cdots,\ n$;
- $x_j^* - \epsilon_j^b \geq 0,\ j = m+1, m+2,\ \cdots,\ n$.

Then, we can construct a new schedule, x^b such that

$$x_j^b = \begin{cases} x_j^* & \text{for} \quad j \in M \backslash \{l\} \\ x_j^* + \epsilon_j^b & \text{for} \quad j = l \\ x_j^* - \epsilon_j^b & \text{for} \quad j = m+1,\ m+2,\ \cdots,\ n. \end{cases}$$

Since $x^{b_l} \leq u_l$, $\sum_{j=1}^{m} x_j^b \leq b_1$ and $\sum_{j=1}^{n} x_j^* = \sum_{j=1}^{n} x_j^b \leq b_2$, x^b is a feasible schedule. Furthermore, since $c_l \leq \min\{c_j |\ j = m+1,\ \cdots,\ n\}$ and $\epsilon_l^b = \sum_{j=m+1}^{n} \epsilon_j^b$,

$$z(x^*) - z(x^b) = -c_l \epsilon_l^b + \sum_{j=m+1}^{n} c_j \epsilon_j^b \geq 0.$$

Thus, x^b is another optimal schedule, and we observe that there exists an optimal schedule, x^* such that $\sum_{j=1}^{m} x_j^* \geq \min\{u_l,\ b_1\}$. Henceforth, without loss of generality, we assume that in an optimal schedule, x^*,

$$\sum_{j=1}^{m} x_j^* \geq \min\{u_l,\ b_1\}. \tag{3}$$

If $x_l^* > \min\{u_l,\ b_1\}$, then x^* is not a feasible schedule. Thus,

$$x_l^* < \min\{u_l,\ b_1\}. \tag{4}$$

By inequalities (2) and (4), there exists a vector $\epsilon^c = (\epsilon_1^c,\ \epsilon_2^c,\ \cdots,\ \epsilon_n^c)$ such that

- $\epsilon_l^c = \min\{u_l,\ b_1\} - x_l^*$;
- $\sum_{j \in M \setminus \{l\}} \epsilon_j^c = \epsilon_l^c$ and $\epsilon_j^c \geq 0,\ j = 1,\ 2,\ \cdots,\ m$;
- $x_j^* - \epsilon_j^c \geq 0$ for $j \in M \setminus \{l\}$.

Then, we can construct a new schedule, x^c such that

$$x_j^c = \begin{cases} x_j^* - \epsilon_j^c & \text{for} \quad j \in M \setminus \{l\} \\ x_j^* + \epsilon_j^c & \text{for} \quad j = l \\ x_j^* & \text{for} \quad j = m+1,\ m+2,\ \cdots,\ n. \end{cases}$$

Since $\sum_{j=1}^{m} x_j^* = \sum_{j=1}^{m} x_j^c$ and $\sum_{j=1}^{n} x_j^* = \sum_{j=1}^{n} x_j^c$, x^c is a feasible schedule. Furthermore, since $c_l = \min\{c_j \mid j = 1,\ 2,\ \cdots,\ m\}$ and $\sum_{j \in M \setminus \{l\}} \epsilon_j^c = \epsilon_l^c$,

$$z(x^*) - z(x^c) = -c_l \epsilon_l^c + \sum_{j \in M \setminus \{l\}} c_j \epsilon_j^c \geq 0.$$

Thus, we can construct another optimal schedule satisfying Lemma 2-*i* from x^* without increasing the objective value.

ii) Suppose that there exists an optimal schedule, x^* such that $x_l^* \neq \min\{u_l,\ b_2\}$. If $\sum_{j=1}^{n} x_j^* < \min\{u_l,\ b_2\}$, then we can construct a new schedule, x^d such that

$$x_j^d = \begin{cases} x_j^* & \text{for} \quad j \in N \setminus \{l\} \\ x_j^* + \min\{u_l,\ b_2\} - \sum_{j=1}^{n} x_j^*, & \text{for} \quad j = l. \end{cases}$$

Since $x_l^d \leq u_l$, $\sum_{j=1}^{m} x_j^* = \sum_{j=1}^{m} x_j^d$ and $\sum_{j=1}^{n} x_j^* \leq \sum_{j=1}^{n} x_j^d \leq b_2$, x^d is a feasible schedule. Furthermore, since $c_l < 0$ and $\min\{u_l,\ b_2\} - \sum_{j=1}^{n} x_j^* > 0$,

$$z(x^*) - z(x^d) = -c_l(\min\{u_l,\ b_2\} - \sum_{j=1}^{n} x_j^*) > 0.$$

This is a contradiction. Thus, we observe that in an optimal schedule, x^*,

$$\sum_{j=1}^{n} x_j^* \geq \min\{u_l,\ b_2\}. \tag{5}$$

If $x_l^* > \min\{u_l,\ b_2\}$, then x^* is not a feasible schedule. Thus,

$$x_l^* < \min\{u_l,\ b_1\}. \tag{6}$$

By inequalities (5) and (6), there exists a vector, $\epsilon^e = (\epsilon_1^e,\ \epsilon_2^e,\ \cdots,\ \epsilon_n^e)$ such that

- $\epsilon_l^e = \min\{u_l,\ b_1\} - x_l^*$;
- $\sum_{j \in N \setminus \{l\}} \epsilon_j^e = \epsilon_l^e$ and $\epsilon_j^e \geq 0,\ j = 1,\ 2,\ \cdots,\ n$;
- $x_j^* - \epsilon_j^e \geq 0,\ j = 1,\ 2,\ \cdots,\ n$.

Then, we can construct a new schedule, x^b such that

$$x_j^e = \begin{cases} x_j^* - \epsilon_j^e & \text{for} \quad j \in N \setminus \{l\} \\ x_j^* + \epsilon_j^e & \text{for} \quad j = l. \end{cases}$$

Since $x_l^e \leq u_l$, $\sum_{j=1}^m x_j^* \geq \sum_{j=1}^m x_j^e$ and $\sum_{j=1}^n x_j^* = \sum_{j=1}^n x_j^e \leq b_2$, x^e is a feasible schedule. If $b_1 = 0$, then it may hold that

$$c_l < \min\{c_j \mid j = 1,\ 2,\ \cdots,\ m\}. \tag{7}$$

However, since $b_1 = 0$, $x_j^* = 0$, $j = 1,\ 2,\ \cdots,\ m$ in x^*, which implies that

$$\epsilon_j^e = 0 \quad \text{for} \quad j = 1,\ 2,\ \cdots,\ m.$$

By equation (7),

$$\sum_{j \in \{m+1,\, m+2,\ \cdots,\, n\} \setminus \{l\}} \epsilon_j^e = \epsilon_l^e. \tag{8}$$

By $c_l = \min\{c_j \mid j = m+1,\ m+2,\ \cdots,\ n\}$ and equation (8),

$$z(x^*) - z(x^e) = -c_l\epsilon_l^e + \sum_{j \in \{m+1,\ m+2, \cdots,\ n\} \setminus \{l\}} c_j \epsilon_j^e \geq 0.$$

If $b_1 > 0$, then $c_l = \min\{c_j \mid j = 1,\ 2,\ \cdots,\ n\}$. Furthermore, since $\sum_{j \in N \setminus \{l\}} \epsilon_j^e = \epsilon_l^e$,

$$z(x^*) - z(x^e) = -c_l\epsilon_l^e + \sum_{j \in N \setminus \{l\}} c_j \epsilon_j^e \geq 0.$$

Thus, we can construct another optimal schedule satisfying Lemma 2-*ii*) from x^* without increasing the objective value. □

<Appendix B>

Proof. i) Suppose that there exists an optimal schedule, x^* such that $x_l^* \neq \min\{u_l,\ b_1,\ a_2\}$. If $\sum_{j=1}^{n} x_j^* < \min\{u_l,\ b_1,\ a_2\}$, then x^* is infeasible. Thus,

$$\sum_{j=1}^{n} x_j^* \geq \min\{u_l,\ b_1,\ a_2\}. \tag{9}$$

If $\sum_{j=1}^{m} x_j^* < \min\{u_l,\ b_1,\ a_2\}$, then, by inequality (9), there exists a vector $\epsilon^a = (\epsilon_1^a,\ \epsilon_2^a,\ \cdots,\ \epsilon_n^a)$ such that

- $\epsilon_l^a = \min\{u_l,\ b_1,\ a_2\} - \sum_{j=1}^{m} x_j^*$;
- $\sum_{j=m+1}^{n} \epsilon_j^a = \epsilon_l^a$ and $\epsilon_j^a \geq 0,\ j = m+1, m+2,\ \cdots,\ n$;
- $x_j^* - \epsilon_j^a \geq 0,\ j = m+1, m+2,\ \cdots,\ n.$

Then, we can construct a new schedule, x^a such that

$$x_j^a = \begin{cases} x_j^* & \text{for} \quad j \in M \setminus \{l\} \\ x_j^* + \epsilon_j^a & \text{for} \quad j = l \\ x_j^* - \epsilon_j^a & \text{for} \quad j = m+1,\ m+2,\ \cdots,\ n. \end{cases}$$

Since $x^{a_l} \leq u_l$, $\sum_{j=1}^{m} x_j^a \leq b_1$ and $\sum_{j=1}^{n} x_j^* = \sum_{j=1}^{n} x_j^a$, x^a is a feasible schedule. Furthermore, since $c_l \leq \min\{c_j \mid j = m+1,\ \cdots,\ n\}$ and $\epsilon_l^a = \sum_{j=m+1}^{n} \epsilon_j^a$,

$$z(x^*) - z(x^a) = -c_l\epsilon_l^a + \sum_{j=m+1}^{n} c_j \epsilon_j^a \geq 0.$$

Thus, x^a is another optimal schedule. Thus, we observe that there exists an optimal schedule, x^* such that $\sum_{j=1}^{m} x_j^* \geq \min\{u_l,\ b_1,\ a_2\}$. Henceforth, without loss of generality, we assume that in an optimal schedule x^*,

$$\sum_{j=1}^{m} x_j^* \geq \min\{u_l,\ b_1,\ a_2\}. \tag{10}$$

If $x_l^* > \min\{u_l,\ b_1,\ a_2\}$, then we know, $a_2 = \min\{u_l,\ b_1,\ a_2\}$. Note that if $\min\{u_l,\ b_1\} = \min\{u_l,\ b_1,\ a_2\}$, then $x_l^* > u_l$ or $x_l^* > b_l$, which implies that x^* is infeasible. Thus, we can construct a new schedule, x^b such that

$$x_j^b = \begin{cases} x_j^* & \text{for} \quad j \in N \setminus \{l\} \\ a_2 & \text{for} \quad j = l. \end{cases}$$

Since $x_l^a = a_2 < u_l$, $\sum_{j=1}^{m} x_j^* \geq \sum_{j=1}^{m} x_j^b$ and $\sum_{j=1}^{n} x_j^* \geq \sum_{j=1}^{n} x_j^b$, x^b is a feasible schedule. Furthermore, since $c_l \geq 0$,

$$z(x^*) - z(x^b) = c_j(x_l^* - a_2) \geq 0.$$

Thus, x^b is another optimal schedule. Thus, we observe that there exists an optimal schedule, x^* such that $x_l^* < \min\{u_l,\ b_1,\ a_2\}$. Henceforth, without loss of generality, we assume that in an optimal schedule, x^*,

$$x_l^* < \min\{u_l,\ b_1,\ a_2\}. \tag{11}$$

By inequalities (10) and (11), there exists a vector, $\epsilon^c = (\epsilon_1^c,\ \epsilon_2^c,\ \cdots,\ \epsilon^{c_n})$ such that

- $\epsilon_l^c = \min\{u_l,\ b_1,\ a_2\} - x_l^*$;
- $\sum_{j \in M \setminus \{l\}} \epsilon_j^c = \epsilon_l^c$ and $\epsilon_j^c \geq 0,\ j = 1,\ 2,\ \cdots,\ m$;
- $x_j^* - \epsilon_j^c \geq 0$ for $j \in M \setminus \{l\}$.

Then, we can construct a new schedule, x^c such that

$$x_j^c = \begin{cases} x_j^* - \epsilon_j^c & \text{for} \quad j \in M \setminus \{l\} \\ x_j^* + \epsilon_j^c & \text{for} \quad j = l \\ x_j^* & \text{for} \quad j = m+1,\ m+2,\ \cdots,\ n. \end{cases}$$

Since $x_l^c \leq u_l$, $\sum_{j=1}^{m} x_j^* = \sum_{j=1}^{m} x_j^c$ and $\sum_{j=1}^{n} x_j^* = \sum_{j=1}^{n} x_j^c$, x^c is a feasible schedule. Furthermore, since $c_l = \min\{c_j \mid j = 1,\ 2,\ \cdots,\ m\}$ and $\sum_{j \in M \setminus \{l\}} \epsilon_j^c = \epsilon_l^c$,

$$z(x^*) - z(x^c) = -c_l\epsilon_l^c + \sum_{j \in M \setminus \{l\}} c_j \epsilon_j^c \geq 0.$$

Thus, we can construct another optimal schedule satisfying Lemma 3-*i*) from x^* without increasing the objective value.

ii) Suppose that there exists an optimal schedule, x^* such that $x_l^* \neq \min\{u_l,\ a_2\}$. If $\sum_{j=1}^{n} x_j^* < \min\{u_l,\ a_2\}$, then x^* is infeasible. Thus,

$$\sum_{j=1}^{n} x_j^* \geq \min\{u_l,\ a_2\}. \tag{12}$$

If $x_l^* > \min\{u_l,\ a_2\}$, then we know $a_2 = \min\{u_l,\ a_2\}$. Note that if $u_l = \min\{u_l,\ a_2\}$, then $x_l^* > u_l$, which implies that x^* is infeasible. Thus, we can construct a new schedule, x^d such that

$$x_j^d = \begin{cases} x_j^* & \text{for} \quad j \in N \setminus \{l\} \\ a_2 & \text{for} \quad j = l. \end{cases}$$

Since $x_l^d = a_2 < u_l$, $\sum_{j=1}^m x_j^* = \sum_{j=1}^m x_j^d$ and $\sum_{j=1}^n x_j^* \geq \sum_{j=1}^n x_j^d$, x^d is a feasible schedule. Furthermore, since $c_l \geq 0$,

$$z(x^*) - z(x^d) = c_j(x_l^* - a_2) \geq 0.$$

Thus, x^d is another optimal schedule. Thus, we observe that there exists an optimal schedule x^* such that $x_l^* < \min\{u_l, a_2\}$. Henceforth, without loss of generality, we assume that in an optimal schedule, x^* such that

$$x_l^* < \min\{u_l, a_2\}. \tag{13}$$

By inequalities (12) and (13), there exists a vector, $\epsilon^e = (\epsilon_1^e, \epsilon_2^e, \cdots, \epsilon_n^e)$ such that

- $\epsilon_l^e = \min\{u_l, a_2\} - x_l^*$;
- $\sum_{j \in N \setminus \{l\}} \epsilon_j^e = \epsilon_l^e$ and $\epsilon_j^e \geq 0$, $j = 1, 2, \cdots, n$;
- $x_j^* - \epsilon_j^e \geq 0$ for $j \in N \setminus \{l\}$.

Then, we can construct a new schedule, x^e such that

$$x_j^e = \begin{cases} x_j^* - \epsilon_j^e & \text{for} \quad j \in N \setminus \{l\} \\ x_j^* + \epsilon_j^e & \text{for} \quad j = l. \end{cases}$$

Since $\sum_{j=1}^m x_j^* \geq \sum_{j=1}^m x_j^e$ and $\sum_{j=1}^n x_j^* = \sum_{j=1}^n x_j^e$, x^e is a feasible schedule. If $b_1 = 0$, then it may hold that

$$c_l < \min\{c_j \mid j = 1, 2, \cdots, m\}.$$

However, since $b_1 = 0$, $x_j^* = 0$, $j = 1, 2, \cdots, m$ in x^*, which implies that

$$\epsilon_j^e = 0 \quad \text{for} \quad j = 1, 2, \cdots, m. \tag{14}$$

By equation (14),

$$\sum_{j \in \{m+1, m+2, \cdots, n\} \setminus \{l\}} \epsilon_j^e = \epsilon_l^e. \tag{15}$$

By $c_l = \min\{c_j \mid j = m+1,\ m+2,\ \cdots,\ n\}$ and equation (15),

$$z(x^*) - z(x^e) = -c_l\epsilon_l^e + \sum_{j \in \{m+1,\ m+2, \cdots,\ n\} \setminus \{l\}} c_j\epsilon_j^e \geq 0.$$

If $b_1 > 0$, then $c_l = \min\{c_j \mid j = 1,\ 2,\ \cdots,\ n\}$. Furthermore, since $\sum_{j \in N \setminus \{l\}} \epsilon_j^e = \epsilon_l^e$,

$$z(x^*) - z(x^e) = -c_l\epsilon_l^e + \sum_{j \in N \setminus \{l\}} c_j\epsilon_j^e \geq 0.$$

Thus, we can construct another optimal schedule satisfying Lemma 3-*ii*) from x^* without increasing the objective value. □