Journal of The Korea Society of Computer and Information
Vol. 23 No. 3, pp. 85-92, March 2018
www.ksci.re.kr
https://doi.org/10.9708/jksci.2018.23.03.085

A time-cost tradeoff problem with multiple interim assessments under the precedence graph with two chains in parallel

Byung-Cheon Choi*, Yunhong Min**

Abstract

We consider a project scheduling problem in which the jobs can be compressed by using additional resource to meet the corresponding due dates, referred to as a time-cost tradeoff problem. The project consists of two independent subprojects of which precedence graph is a chain. The due dates of jobs constituting the project can be interpreted as the multiple assessments in the life of project. The penalty cost occurs from the tardiness of the job, while it may be avoided through the compression of some jobs which requires an additional cost. The objective is to find the amount of compression that minimizes the total tardy penalty and compression costs. Firstly, we show that the problem can be decomposed into several subproblems whose number is bounded by the polynomial function in n, where n is the total number of jobs. Then, we prove that the problem can be solved in polynomial time by developing the efficient approach to obtain an optimal schedule for each subproblem.

▸Keyword: Project scheduling, Time-cost tradeoff, Parallel precedence graph

I. Introduction

In project management, a manager need to consider two conflicting objectives: (1) minimizing project completion time and (2) minimizing the project cost. The completion time, referred to as a makespan, of a project can be reduced by investing more resources which incurs the increase of project cost. The time-cost tradeoff problem (TCTP) has a form either of minimization of the project's cost under a specified due date or of minimization of makespan under a given budget.

When the makespan of a project is considered either as an objective or a constraint, we assume that only one assessment exists during the whole project. In reality, however, there may exist multiple assessments in the middle of the project with respect to the corresponding interim due dates, and thus some penalties can occur if the project at each assessment does not go along according to a planned schedule. For example, a venture capital company begins to makes small investments on a start-up and then determines whether the additional investment is carried out or not, based on what the start-up achieves compared with its initial plan [3, 14].

In this paper, we consider the TCTP with multiple assessments in which a project consists of multiple jobs and some jobs have their own due dates. Jobs with their own due dates are called milestones. In general, there exist the precedence relations among jobs and these relations are represented by the graph, called the precedence graph. In the precedence graph, each node corresponds to job and the arc between two jobs represents their precedence.

• First Author: Byung-Cheon Choi, Corresponding Author: Yunhong Min
*Byung-Cheon Choi (polytime@cnu.ac.kr), School of buisiness, Chungnam National University
**Yunhong Min (yunhong.min@inu.ac.kr), Graduate School of Logistics, Incheon National University
• Received: 2018. 01. 30, Revised: 2018. 02. 05, Accepted: 2018. 02. 28.
• This work was supported by research fund of Chungnam National University in 2017.

Each job has its own processing time which can be compressed through the use of some resources, e.g., human or capital. If a milestone fails to be completed within its due date, it incurs the penalty cost. The penalty cost, however, can be avoided by reducing or compressing the processing time of some jobs. The objective is to minimize the sum of the tardiness and the compression costs. We describe the tardy and compression costs as a weight of the tardy milestone and the linear function for the amount of compression, respectively.

Our project scheduling problem belongs to the class of the TCTP with the linear compression cost function, referred to as a LTCTP. For a comprehensive review of the TCTP with more general compression cost function (e.g., convex or concave), see [2, 10, 12, 13]. Fulkerson [9] and Kelley [11] considered the LTCTP to minimize the makespan, subject to a constraint on the budget for the total amount of compression, and proposed an efficient algorithm based on the network flow model. Choi and Chung [5] considered two LTCTP's with multiple milestones on a single chain precedence graph. The objective of the first is to minimize the total weighted number of tardy jobs with a constraint on the total compression cost. The second objective is to minimize the sum of the total compression cost and the total weighted number of tardy jobs. They proved the weak NP-hardness of the first problem and the polynomiality of the second. Choi and Park [6] considered the general version of the second problem in [5] such that the compression cost function is convex or concave. They proved the weak NP-hardness and the polynomiality of the problems with the concave and the convex compression cost functions, respectively. Choi and Chung [7] considered the problem in [6] with the concave compression cost function, and investigated the optimality properties that make the problem polynomially solvable. The problem in this paper can be considered as the general version of the problem in [5, 6, 7], in that more general precedence graph, i.e., two chains in parallel, is considered. The precedence graph of two chains in parallel is motivated from the situation such that

The project consists of several independent subproject whose precedence graph can be described as a chain.

The project is started after dividing up the project into several subproejcts, and completed after assembling the completed subprojects.

Our problem can be formally stated as follows. The problem can be represented by a activity-on-node graph $G=(V,E)$ consisting of two chains in parallel, where $V=\{(i,j)|i=0,1,2, j=1,2,...n_i\}$ is the set of jobs and E is the set of precedence relations. Then, the resulting precedence graph is described as Fig. 1. Let the jobs in $J_0=\{(0,1),(0,2)\}$ be referred to as the set of start and terminal jobs.

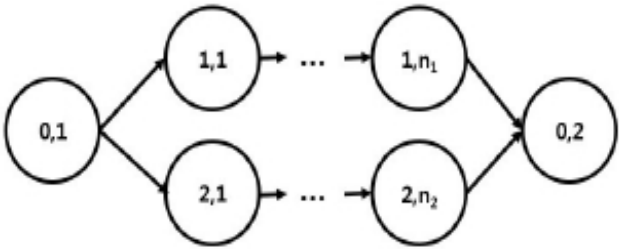

Fig. 1. The precedence graph

For the chain $i=1,2$, $J_i=\{(i,j)|j=1,2,...n_i\}$ is the set of jobs consisting the chain i. Associated with job $(i,j)\in J_i$, is an initial processing time $p_{i,j}$, a maximal amount for compression $u_{i,j}$ and a compression cost ratio $c_{i,j}$. Let $x=(x_{i,j})_{(i,j)\in V}$ be the vector of which component $x_{i,j}$ is the compression amount of job (i,j) subject to $0\le x_{i,j}\le u_{i,j}$. Note that $x=(x_{i,j})_{(i,j)\in V}$ completely characterizes the schedule, because there exists an optimal schedule such that each job is started as soon as possible. Thus, we call $x=(x_{i,j})_{(i,j)\in V}$ a schedule. Let job (i,j) be uncompressed and fully compressed in a schedule x, if $x_{i,j}=0$ and $x_{i,j}=u_{i,j}$, respectively. Let $D\subseteq V$ be the set of milestones, i.e., jobs with due dates. For $(i,j)\in D$, let $d_{i,j}$ and $w_{i,j}$ be the due date and the penalty cost for tardiness of milestone (i,j), respectively. Let $C_{i,j}(x)$ be the completion time of job (i,j) in a schedule x. Note that we can transform the case with $D\subseteq V$ into the one with $D=V$ by letting $d_{i,j}=\sum_{(i,j)\in V} p_{i,j}$ and $w_{i,j}=0$ for $(i,j)\in V-D$. It is clear that the optimal schedules of the original and transformed cases are identical. Without loss of generality, henceforth, we assume that $D=V$, and call milestone as job. Our problem, called Problem P, can be formulated as follows:

$$\min z(x)=\sum_{(i,j)\in T(x)} w_{i,j}+\sum_{(i,j)\in V} c_{i,j}x_{i,j}$$

$$s.t.\ C_{i,j}(x)+p_{i',j'}-x_{i',j'}\le C_{i',j'}(x),((i,j),(i',j'))\in E$$

$$0\le x_{i,j}\le u_{i,j}, i=0,1,2, j=1,2,...,n_i,$$

where $T(x)$ is the set of tardy jobs in x. To exclude a trivial case, we assume that the zero vector x^0 with $x^0_{i,j}=0$ for each $(i,j)\in V$ is not an optimal schedule.

The remainders of the paper are organized as follows. Section 2 decomposes Problem P into the smaller subproblems whose polynomialities imply the polynomiality of Problem P. Section 3 develops the effi cient algorithms for each subproblem. Finally, Section 4 presents some concluding remarks and future works.

II. Decomposition of Problem P

In this section, we show that Problem P can be decomposed into the subproblems by using the concept of the just-in-time (JIT) job. A job (i,j) is said to be just-in-time (JIT), if it is completed exactly at $d_{i,j}$ in an optimal schedule. Firstly, we present some optimality conditions.

Lemma 1 There exists an optimal schedule such that at least one JIT job exists.

Proof. Suppose that a job (i,j) is compressed and the JIT job does not exist in an optimal schedule x^*. We can construct a new schedule $\bar{x}$ by letting $\bar{x}_{ij}=x^*_{i,j}-\varepsilon$ where $\varepsilon>0$ is sufficiently small value. Since

$$T(\bar{x})=T(x^*) \text{ and } \sum_{(i,j)\in V} c_{i,j}\bar{x}_{i,j} < \sum_{(i,j)\in V} c_{i,j}x^*_{i,j},$$

however, $z(\bar{x})<z(x^*)$. This is a contradiction. ■

Lemma 2 There exists an optimal schedule such that the jobs after the last JIT jobs in $J_0\cup J_1$ and $J_0\cup J_2$ are uncompressed.

Proof. If job (0,2) is the JIT job in an optimal schedule x^*. Then, we assume that job (0,2) is not the JIT job in x^*. Suppose that in x^* there exists a compressed job (i,j) after the last JIT jobs in $J_0\cup J_1$ or $J_0\cup J_2$. Then, we can construct a new schedule $\bar{x}$ by letting $\bar{x}_{ij}=x^*_{i,j}-\varepsilon$ where $\varepsilon>0$ is sufficiently small value. Since

$$T(\bar{x})=T(x^*) \text{ and } \sum_{(i,j)\in V} c_{i,j}\bar{x}_{i,j} < \sum_{(i,j)\in V} c_{i,j}x^*_{i,j}$$

however, $z(\bar{x})<z(x^*)$. This is a contradiction. ■

For each $i=1,2$, let (i,a_i) and (i,b_i) be the first and the last JIT jobs on chain i in the optimal schedule, respectively. Depending on the existences of jobs (i,a_i) and (i,b_i) on each chain i, the structure of the optimal schedule belongs to one of the three cases in Fig. 2. Note that if chain i has one JIT job, then

$a_i=b_i$ for each $i=1,2$.

In Case 1, both chains have the JIT job, while in Cases 2 and 3, either one and neither of chains has no JIT job, respectively.

It is observed that if we know the existences and positions of JIT jobs in an optimal solution, the schedules of the remaining jobs in the optimal solution can be obtained by solving some of subproblems in Fig. 3. For example, Cases 1 and 2 consist of subproblems 1-3 and subproblems 1 and 5, respectively.

By the observation above, we can have the following theorem.

Theorem 1 Problem P can be solved in polynomial time if the five subproblems in Fig. 3 are polynomially solvable.

Proof. It is observed that the possible number of the cases is $O(n_1^2n_2^2)$ depending on which jobs in each chain become the first and the last JIT job. Since each case belongs to one type of Cases 1-3 and consists of at most four subproblems in Fig. 3, the total number of the subproblems that we should solve to find an optimal schedule for Problem P is $O(n_1^2n_2^2)$. Henceforth, we show that Problem P is polynomially solvable by proving the polynomiality of Subproblems 1-5. ■

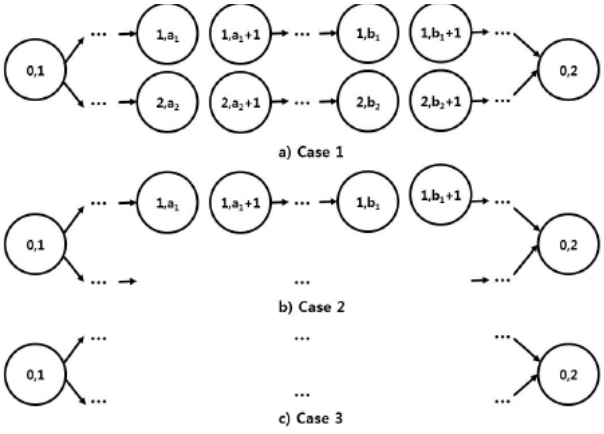

Fig. 2. Three cases

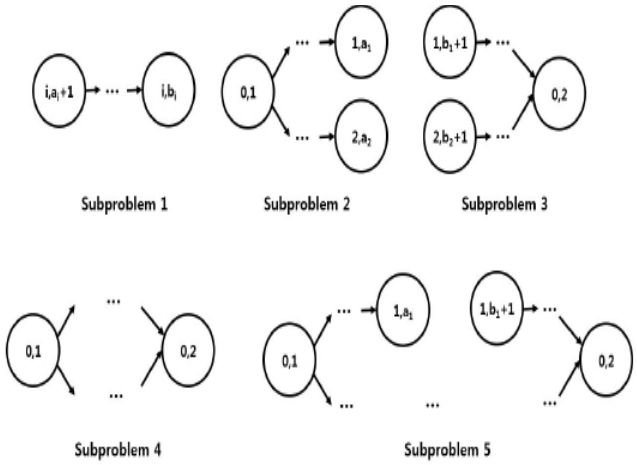

Fig. 3. Five subproblems

III. Polynomiality of Subproblems

In this section, we show that five subproblems in Fig. 3 are polynomially solvable. It is known from [5] that the polynomial solvability of Subproblem 1 was proved. Thus, we consider only Subproblems 2–5. The following notation is useful for the subsequent expositions. For each chain $i=1,2$,

$A_i=\{(i,j)|j=1,2,...a_i\}$ and $B_i=\{(i,j)|j=b_i+1,...,n_i\}$

1. Subproblem 2

For simplicity, we introduce some notation. For each chain $i=1,2$, let

$\Delta_{\min}=\min\{\Delta_1,\Delta_2\}$ and $\Delta_{\max}=\max\{\Delta_1,\Delta_2\}$,

where $\Delta_i=\sum_{(i,j)\in\{(0,1)\}\cup A_i} p_{i,j}-d_{i,a_i}$ and

$l_i=\arg min\{c_{i,j}|u_{i,j}>0 \text{ for } (i,j)\in A_i\}$ (1)

If multiple jobs have the minimum compression cost ratio in (1), then the one processed earliest is selected, because this way guarantees the schedule with the smallest total penalty for the tardy jobs.

Lemma 3 There exists an optimal schedule x^* for Subproblem 2 such that

(i) If $\Delta_{\max}>\Delta_{\min}$ then,

$x^*_{i,l_i}>\min\{\Delta_i-\Delta_{\min},u_{i,l_i}\}$ for $i=1,2$.

(ii) If $\Delta_{\max}=\Delta_{\min}$ and $c_{0,1}\le c_{1,l_1}+c_{2,l_2}$ then,

$x^*_{0,1}=\min\{\Delta_{\min},u_{0,1}\}$.

(iii) If $\Delta_{\max}=\Delta_{\min}$ and $c_{0,1}>c_{1,l_1}+c_{2,l_2}$ then,

$x^*_{i,l_i}\ge\min\{\Delta_{\min},u_{1,l_1},u_{2,l_2}\}$ for $i=1,2$.

Proof. Firstly, consider the case (i). Without loss of generality, assume that $\Delta_1=\Delta_{\max}$ and $\Delta_2=\Delta_{\min}$. Suppose that $x^*_{1,l_1}<\min\{\Delta_1-\Delta_2,u_{1,l_1}\}$. This implies the existence of job $(1,\alpha_1)$ with $c_{1,l_1}\le c_{1,\alpha_1}$ and $x_{1,\alpha_1}>0$. If $c_{1,l_1}<c_{1,\alpha_1}$ then we can construct a new schedule $\bar{x}$ by letting $\bar{x}_{1,l_1}=x^*_{1,l_1}+\varepsilon$ and $\bar{x}_{1,\alpha_1}=x^*_{1,\alpha_1}-\varepsilon$ where $\varepsilon>0$ is sufficiently small. Since $z(\bar{x})<z(x^*)$, this is a contradiction. If $c_{1,l_1}=c_{1,\alpha_1}$, then we can construct a new schedule $\bar{x}$ by letting $\bar{x}_{1,l_1}=x^*_{1,l_1}+\theta$ and $\bar{x}_{1,\alpha_1}=x^*_{1,\alpha_1}-\theta$ where $\theta=\min\{u_{1,l_1}-x^*_{1,l_1},x^*_{1,\alpha_1}\}$. Since job $(1,l_1)$ is processed before job $(1,\alpha_1)$, $z(\bar{x})<z(x^*)$. By applying this argument repeatedly, we can construct another optimal schedule satisfying (i) or a contradiction occurs. The cases (ii) and (iii) can be proved by the similar argument, and hence we omit them. ■

Based on Lemma 3, we can construct an algorithm to obtain an optimal schedule for Subproblem 2 as follows:

Algorithm Out-Tree

Step 1. Set $x_{0,j}=0$ and $x_{i,j}=0$ for $i=1,2$ and $j=1,2,...,a_i$.

Step 2. Find $\{(1,l_1),(2,l_2)\}$.

Step 3. If $\Delta_{\max}>\Delta_{\min}$ then go to Step 3-1, while go to Step 3-2, otherwise.

Step 3-1 For $i=1,2$, update $x_{i,l_i},u_{i,l_i},p_{i,l_i}$ and $\Delta_{\max}$ by letting

$x_{i,l_i}=x_{i,l_i}+\bar{\Delta}_i$, $u_{i,l_i}=u_{i,l_i}-\bar{\Delta}_i$, $p_{i,l_i}=p_{i,l_i}-\bar{\Delta}_i$ and

$\Delta_{\max}=\Delta_{\max}-\max\{\bar{\Delta}_1,\bar{\Delta}_2\}$,

where $\bar{\Delta}_i=\min\{\Delta_i-\Delta_{\min},u_{i,l_i}\}$, and go to Step 3-3

Step 3-2 If $c_{0,1}\le c_{1,l_1}+c_{2,l_2}$, then update

$x_{0,1},u_{0,1},p_{0,1}$ and $\Delta_{\min}$ by letting

$x_{0,1}=x_{0,1}+\hat{\Delta}$, $u_{0,1}=u_{0,1}-\hat{\Delta}$, $p_{0,1}=p_{0,1}-\hat{\Delta}$ and

$\Delta_{\min}=\Delta_{\min}-\hat{\Delta}$,

where $\hat{\Delta}=\min\{\Delta_{\min},u_{0,1}\}$.

Otherwise, for $i=1,2$, update $x_{i,l_i},u_{i,l_i},p_{i,l_i}$ and $\Delta_{\min}$ by letting

$x_{i,l_i}=x_{i,l_i}+\tilde{\Delta}$, $u_{i,l_i}=u_{i,l_i}-\tilde{\Delta}$, $p_{i,l_i}=p_{i,l_i}-\tilde{\Delta}$ and

$\Delta_{\min}=\Delta_{\min}-\tilde{\Delta}$,

where $\tilde{\Delta}=\min\{\Delta_{\min},u_{1,l_1},u_{2,l_2}\}$.

Step 3-3 If $\Delta_{\min}=0$, then STOP.

Step 3-4 If $u_{0,1}=0$, then set $c_{0,1}=\infty$ and go to Step 2.

Note that Algorithm Out-Tree runs in $O(n)$.

Theorem 2 Subproblem 2 can be solved in polynomial time.

Proof. It holds immediately from Algorithm Out-Tree. ■

2. Subproblem 3

It is observed from Lemma 2 that if job (0,2) is not a JIT job, then all jobs are uncompressed in an optimal schedule for Subproblem 3. Thus, we assume that job (0,2) is a JIT job in the optimal schedule. For chain $i=1,2$, let

$\delta_{\min}=m\{\delta_1,\delta_2\}$ and $\delta_{\max}=\max\{\delta_1,\delta_2\}$,

where $\delta_i=\max\left\{0,d_{i,b_i}+\sum_{(i,j)\in\{(0,2)\}\cup B_i}p_{i,j}-d_{0,2}\right\}$ and

$l_i=\arg min\{c_{i,j}|u_{i,j}>0 \text{ for } (i,j)\in B_i\}$ (2)

If multiple jobs have the minimum compression cost ratio in (2), then the one processed earliest is selected, because this way guarantees the schedule with the smallest total penalty. Then, we can characterize the properties of the optimal schedule of which statements and proofs are similar to Lemma 3 (Thus, we omit the proof).

Lemma 4 There exists an optimal schedule x^* for Subproblem 3 such that

(i) If $\delta_{\max}>\delta_{\min}$ then,

$x^*_{i,l_i}>\min\{\delta_i-\delta_{\min},u_{i,l_i}\}$ for $i=1,2$.

(ii) If $\delta_{\max}=\delta_{\min}$ and $c_{0,2}\le c_{1,l_1}+c_{2,l_2}$ then,

$x^*_{0,2}=\min\{\delta_{\min},u_{0,2}\}$.

(iii) If $\delta_{\max}=\delta_{\min}$ and $c_{0,2}>c_{1,l_1}+c_{2,l_2}$ then,

$x^*_{i,l_i}\ge\min\{\delta_{\min},u_{1,l_1},u_{2,l_2}\}$ for $i=1,2$.

Based on Lemma 4, we cam construct an algorithm to obtain an optimal schedule for Subproblem 3 as follows:

Algorithm In-Tree

Step 1. Set $x_{0,2}=0$ and $x_{i,j}=0$ for $i=1,2$ and $j=b_i+1,b_i+2,...,n_i$.

Step 2. Find $\{(1,l_1),(2,l_2)\}$.

Step 3. If $\delta_{\max}>\delta_{\min}$ then go to Step 3-1, while go to Step 3-2, otherwise.

Step 3-1 For $i=1,2$, update $x_{i,l_i},u_{i,l_i},p_{i,l_i}$ and $\delta_{\max}$ by letting

$x_{i,l_i}=x_{i,l_i}+\bar{\delta}_i$, $u_{i,l_i}=u_{i,l_i}-\bar{\delta}_i$, $p_{i,l_i}=p_{i,l_i}-\bar{\delta}_i$ and

$\delta_{\max}=\delta_{\max}-\max\{\bar{\delta}_1,\bar{\delta}_2\}$,

where $\bar{\delta}_i=\min\{\delta_i-\delta_{\min},u_{i,l_i}\}$, and go to Step 3-3.

Step 3-2 If $c_{0,2}\le c_{1,l_1}+c_{2,l_2}$, then update

$x_{0,2},u_{0,2},p_{0,2}$ and $\delta_{\min}$ by letting

$x_{0,2}=x_{0,2}+\hat{\delta}$, $u_{0,2}=u_{0,2}-\hat{\delta}$, $p_{0,2}=p_{0,2}-\hat{\delta}$ and

$\delta_{\min}=\delta_{\min}-\hat{\delta}$,

where $\hat{\delta}=\min\{\delta_{\min},u_{0,2}\}$. Otherwise, for $i=1,2$, update $x_{i,l_i},u_{i,l_i},p_{i,l_i}$ and $\delta_{\min}$ by letting

$x_{i,l_i}=x_{i,l_i}+\tilde{\delta}$, $u_{i,l_i}=u_{i,l_i}-\tilde{\delta}$, $p_{i,l_i}=p_{i,l_i}-\tilde{\delta}$ and

$\delta_{\min}=\delta_{\min}-\tilde{\delta}$,

where $\tilde{\delta}=\min\{\delta_{\min},u_{1,l_1},u_{2,l_2}\}$.

Step 3-3 If $\delta_{\min}=0$, then STOP.

Step 3-4 If $u_{0,2}=0$, then set $c_{0,2}=\infty$ and go to Step 2.

Note that Algorithm In-Tree runs in $O(n)$.

Theorem 3 Subproblem 3 can be solved in polynomial time.

Proof. It holds immediately from Algorithm In-Tree. ■

3. Subproblem 4

In this subsection, we show that Subproblem 4 can be reduced to either Subproblem 2 or Subproblem 3..

Lemma 5 There exists an optimal schedule x^* for Subproblem 4 such that at least one of jobs in J_0 is fully compressed or uncompressed.

Proof. Suppose that each job in J_0 is partially compressed in x^*. Consider two cases.

(i) $c_{0,1}\ne c_{0,2}$: For simplicity, let $c_{0,1}<c_{0,2}$. We can construct a new schedule $\bar{x}$ by letting $\bar{x}_{0,1}=x^*_{0,1}+\varepsilon$ and $\bar{x}_{0,2}=x^*_{0,2}-\varepsilon$ where $\varepsilon>0$ is sufficiently small value. Then, it is observed that

$T(\bar{x})=T(x^*)$ and $\sum_{(i,j)\in V}c_{i,j}\bar{x}_{i,j}<\sum_{(i,j)\in V}c_{i,j}x^*_{i,j}$

which implies $z(\bar{x})<z(x^*)$.

(ii) $c_{0,1}=c_{0,2}$: We can construct a new schedule $\bar{x}$ by letting $\bar{x}_{0,1}=x^*_{0,1}+\theta$ and $\bar{x}_{0,2}=x^*_{0,2}-\theta$, where $\theta=\min\{u_{0,1}-x^*_{0,1}, x^*_{0,2}\}$. It is observed that

$T(\bar{x}) \le T(x^*)$ and $\sum_{(i,j)\in V} c_{i,j}\bar{x}_{i,j} = \sum_{(i,j)\in V} c_{i,j}x^*_{i,j}$

which implies $z(\bar{x}) \le z(x^*)$. ■

Theorem 4 Subproblem 4 can be solved in polynomial time.

Proof. By Lemma 5, it suffices to consider the following two cases

(i) Job (0,1) is fully compressed or uncompressed: In this case, Subproblem 4 is reduced to Subproblem 3 with new jobs (0,3) and (0,4) such that the processing times of jobs (0,3) and (0,4) are $p_{0,1}$ or $p_{0,1}-u_{0,1}$, and their maximal amount for compressions are zeros (See Fig. 4).

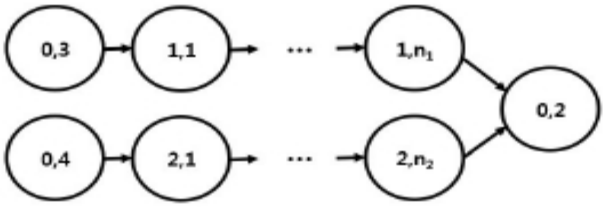

Fig. 4. The precedence graph for Subproblem 3 reduced from Subproblem 4

(ii) Job (0,2) is fully compressed or uncompressed: If job (0,2) is not a JIT job, then it is observed from Lemma 2 that in an optimal schedule x^*,

$x^*_{i,j}=0$ for $(i,j)\in J_1\cup J_2\cup\{(0,2)\}$.

Thus,

$$x^*_{0,1}=\begin{cases}\max\{0,\Delta_{0,1}\} & \text{if } c_{0,1}\max\{0,\Delta_{0,1}\}\le w_{0,1} \text{ and } \Delta_{0,1}\le u_{0,1}\\ 0 & otherwise\end{cases}$$

where $\Delta_{0,1}=p_{0,1}-d_{0,1}$. If job (0,2) is a JIT job, then Subproblem 4 is reduced to Subproblem 2 with new JIT jobs (0,5) and (0,6) such that the processing times of jobs (0,5) and (0,6) are $p_{0,2}$ or $p_{0,2}-u_{0,2}$ and their maximal amount for compressions are zeros (See Fig. 5). ■

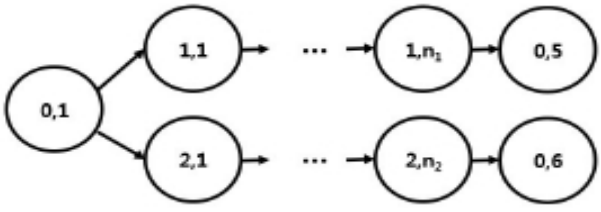

Fig. 5. The precedence graph for Subproblem 2 reduced from Subproblem 4

4. Subproblem 5

For Subproblem 5, it is observed from Lemma 2 that if job (0,2) is not a JIT job, all jobs in $J_2\cup B_1\cup\{(0,2)\}$ are uncompressed in an optimal schedule, which implies that Subproblem 5 is reduced to Subproblem 1. Thus, to exclude this case, throughout this subsection we assume that job (0,2) is a JIT job.

Lemma 6 If at least one of jobs in J_0 is fully compressed or uncompressed in an optimal schedule, then Subproblem 5 is decomposed into some of Subproblems 1-3.

Proof. Consider two cases.

(i) Job (0,1) is fully compressed or uncompressed: Subproblem 5 is reduced to Subproblems 1 and 3 with new jobs (0,3) and (0,4) such that the processing imes of jobs (0,3) and (0,4) are $p_{0,1}$ or $p_{0,1}-u_{0,1}$, and their maximal amount for compressions are zeros (See Fig. 6).

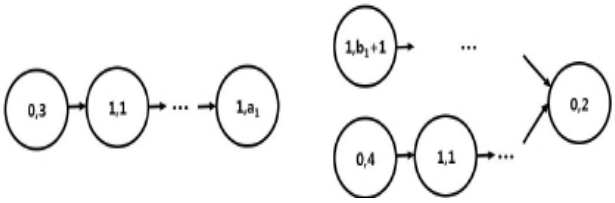

Fig. 6. The precedence graphs for Subproblem 1 and 3 reduced from Subproblem 5

(ii) Job (0,2) is fully compressed or uncompressed: Subproblem 5 is reduced to Subproblems 1 and 2 with new JIT jobs (0,5) and (0,6) such that the processing times of jobs (0,5) and (0,6) are $p_{0,2}$ or $p_{0,2}-u_{0,2}$, and their maximal amount for compressions are zeros (See Fig. 7) ■

Lemma 7 Suppose that jobs (0,1) and (0,2) are partially compressed in an optimal schedule x^*. Then, there exists an optimal schedule x^* for Subproblem 5 such that all jobs in A_1 or B_1 are fully compressed or uncompressed.

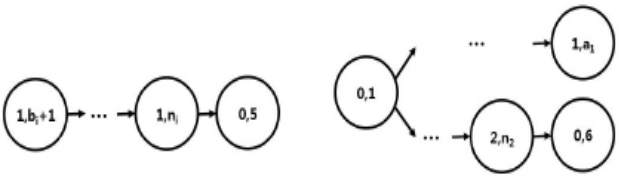

Fig. 7. The precedence graphs for Subproblems 1 and 2 reduced from Subproblem 5

Proof. Suppose that jobs $(1,\alpha)\in A_1$ and $(1,\beta)\in B_1$ are partially compressed in x^*. Consider two cases.

(i) $c_{0,1}+c_{1,\beta}\neq c_{1,\alpha}+c_{0,2}$: For simplicity, let $c_{0,1}+c_{1,\beta}<c_{1,\alpha}+c_{0,2}$. Then, we can construct a new schedule $\bar{x}$ by letting $\bar{x}_{0,1}=x^*_{0,1}+\varepsilon$, $\bar{x}_{1,\beta}=x^*_{1,\beta}+\varepsilon$, $\bar{x}_{0,2}=x^*_{0,2}-\varepsilon$, and $\bar{x}_{1,\alpha}=x^*_{1,\alpha}-\varepsilon$ where $\varepsilon>0$ is sufficiently small value. Then, it is observed that

$T(\bar{x})=T(x^*)$ and $\sum_{(i,j)\in V} c_{i,j}\bar{x}_{i,j}<\sum_{(i,j)\in V} c_{i,j}x^*_{i,j}$

which implies $z(\bar{x})<z(x^*)$.

(ii) $c_{0,1}+c_{1,\beta}=c_{1,\alpha}+c_{0,2}$: We can construct a new schedule $\bar{x}$ by letting $\bar{x}_{0,1}=x^*_{0,1}+\theta$, $\bar{x}_{1,\beta}=x^*_{1,\beta}+\theta$, $\bar{x}_{0,2}=x^*_{0,2}-\theta$, and $\bar{x}_{1,\alpha}=x^*_{1,\alpha}-\theta$ where $\theta=\min\{u_{0,1}-x^*_{0,1}, u_{1,\beta}-x^*_{1,\beta}, x^*_{0,2}, x^*_{1,\alpha}\}$. It is observed that

$T(\bar{x})\le T(x^*)$ and $\sum_{(i,j)\in V} c_{i,j}\bar{x}_{i,j}=\sum_{(i,j)\in V} c_{i,j}x^*_{i,j}$

which implies $z(\bar{x})\le z(x^*)$. ■

Lemma 8 Suppose that jobs (0,1) and (0,2) are partially compressed in an optimal schedule x^*. If a partially compressed job exists in A_1 for x^*, then, Subproblem 5 is reduced to Subproblem 2, while it is reduced to Subproblem 3, otherwise.

Proof. Consider two cases.

(i) A partially compressed job exists in A_1 for x^*: By Lemma 7, this case implies that the jobs in B_1 are fully compressed or uncompressed in x^*. Let

$\bar{\theta}_{\max}=\arg max\{c_{i,j}\,|\,x^*_{i,j}=u_{i,j} \text{ for } (i,j)\in B_1\}$

and

$\bar{\theta}_{\min}=\arg min\{c_{i,j}\,|\,x^*_{i,j}=0 \text{ for } (i,j)\in B_1\}$

Then, it is observed that $(x^*_{i,j})_{(i,j)\in B_1}$ becomes the one of the vectors such that

Some jobs are fully compressed and the others are uncompressed;

$c_{1,\bar{\theta}_{\max}}\le c_{1,\bar{\theta}_{\min}}$;

If $c_{1,\bar{\theta}_{\max}}=c_{1,\bar{\theta}_{\min}}$, job $(1,\bar{\theta}_{\max})$ is processed before job $(1,\bar{\theta}_{\min})$.

For example, assume that $c_{1,1}\le\ldots\le c_{1,a_1}$. Then, the set of these vectors is

$\{(0,0,\ldots,0),(u_{1,b_1+1},0,\ldots,0),\ldots,(u_{1,b_1+1},\ldots,u_{1,n_1})\}$

Note that the value of $x^*_{0,2}$ corresponding to each vector can be determined because job (0,2) is a JIT job. The precedence graph corresponding to each vector is the second graph in Fig. 7.

(ii) A partially compressed job exists in B_1 for x^*: By Lemma 7, this case implies that the jobs in A_1 are fully compressed or uncompressed in x^*. Thus this case is similarly proved by changing the role of A_1 and B_1 in e case (i). The precedence graph corresponding to this case is the second graph in Fig. 6. ■

Theorem 5 Subproblem 5 can be solved in polynomial time.

Proof. It holds immediately from Theorems 2-3 and Lemmas 6-8. ■

IV. Conclusions

We consider the LTCTP with the multiple interim assessments, when the precedence graph is two chains in parallel. The objective is to minimize the total weighted number of tardy jobs plus the total compression cost. Firstly, we decompose the problem into subproblems, based on JIT jobs whose number is bounded by the total number of jobs. Then, we show that our problem is polynomially solvable by proving the polynomiality of each subproblem.

For future research, it would be interesting to consider the case such that the precedence graph consists of more than two chains in parallel, or the compression cost function is convex or concave function.

REFERENCES

[1] C. Artigues, D. Demassey., E. Neron. *"Resource-Constrained Project Scheduling," Wiley*, 2008.

[2] E.B. Berman. "Resource allocation in a PERT network under continuous activity time-cost function," Management Science Vol. 10, pp. 734-745, July 1964.

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

Jan. 2000.
[4] P. Bruker, A. Drexl, R. Mohring, K. Neumann, and E. Perch. “Resource-constrained project scheduling: Notations, classfication, models and methods,” European Journal of Operational Research, Vol. 112, pp. 3-41, Jan. 1999.
[5] B.C. Choi, and J.B. Chung. “Complexity results for linear time-cost tradeoff problem with multiple milestones and completely ordered jobs,” European Journal of Operational Research, Vol. 236, pp. 61-68, July 2014.
[6] B.C. Choi, and M.J. Park. “A continuous time-cost tradeoff problem with multiple milestones and completely ordered jobs,” European Journal of Operational Research Vol. 244, pp. 748-752, Aug. 2015.
[7] B.C. Choi, and J.B. Chung. “Some special cases of a continuous time-cost tradeoff problem with multiple milestones under a chain precedence graph,” Management Science and Financial Engineering, Vol. 22, pp. 5-12, May 2016.
[8] E.L. Demeulemeester and W.S. Herroelen. “Project scheduling - A research handbook,” Kluwer Academic, 2002.
[9] D.R. Fulkerson. “A network flow computation for project cost curves,” Management Science, Vol. 7, pp. 167-178, Jan. 1961.
[10] J.E. Falk, and J.L. Horowitz. “Critical path problems with concave cost-time curves,” Management Science, Vol. 19, pp. 446-455, Dec. 1972.
[11] J.E. Kelley. “Critical path planning and scheduling: Mathematical basis,” Operations Research, Vol. 9, pp. 296-320, June 1961.
[12] L.R. Lamberson, and R.R. Hocking. “Optimum time compression in project scheduling,” Management Science, Vol. 16, pp. 597-606, June 1970.
[13] J. Moussourakis, and C. Haksever. “Project compression with nonlinear cost functions,” Journal of Construction Engineering and Management, Vol. 136, pp. 251-[259, Feb. 2010.
[14] W.A. Sahlmann. “Insights from the ventures capital model of project governance,” Business Economics, Vol. 29, pp. 35-41, July 1994.
[15] J. Weglarz. *“Project scheduling - Recent models algorithms and applications,”* Kluwer Academic, 1999.
[16] J. Weglarz, 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, pp. 177-205, Feb. 2011.

Authors

Byung-Cheon Choi received the B.S. and Ph.D. in Industrial Engineering from Seoul National University in 2000 and 2006, respectively. Dr. Choi joined the faculty of School of Business at Chungnam National University in 2009. He is currently an professor in School of Business at Chungnam National University. He is interested in combinatorial optimization, scheduling theory, and operations management.

Yunhong Min received the B.S. degree in Industrial Engineering & Management Science from POSTECH, and Ph.D. in Industrial Engineering from Seoul National University in 2006 and 2012, respectively. Dr. Min joined the faculty of Graduate School of Logistics at Incheon National University in 2017. Before joining Incheon National University, he was a research staff member of Samsung Advanced Institute of Technology. He is interested in mathematical optimization and machine learning and their applications to logistics and supply chain management.