Journal of Computations & Modelling, vol.1, no.1, 2011, 31-47
ISSN: 1792-7625 (print), 1792-8850 (online)
International Scientific Press, 2011

Resolution of Resource Conflict for a Single Project in Max-Plus Linear Representation

Shotaro Yoshida[1], Hirotaka Takahashi[2] and Hiroyuki Goto[3]

Abstract

This research proposes a method for resolving resource conflicts for a single project. We focus on the framework of resolving resource conflicts in the Critical Chain Project Management (CCPM), which is expressed in the form of Max-Plus Linear (MPL) system. Although the previous studies proposed a method for inserting time buffers, the problem of resource conflict has not been considered for simplicity. On the other hand, the proposed method can detect resource conflicts by checking if time lines of processes overlap. If a resource conflict is detected, the process with low priority is moved up. For moving up the schedule, we define a new adjacency matrix, by which we can resolve the conflict.

Keywords: Max-Plus Algebra, Max-Plus Linear System, Critical Chain Project Management, Resource Conflict

[1] Department of Management and Information Systems Engineering, Nagaoka University of Technology, Nagaoka, Niigata 940-2188, Japan, e-mail: s083381@stn.nagaokaut.ac.jp
[2] Department of Humanities, Yamanashi Eiwa College, Kofu, Yamanashi 400-8555, Japan, e-mail: hirotaka@yamanashi-eiwa.ac.jp
[3] Department of Management and Information Systems Science, Nagaoka University of Technology, Nagaoka, Niigata 940-2188, Japan, e-mail: hgoto@kjs.nagaokaut.ac.jp

Article Info: *Revised* : July 17, 2011. *Published online* : August 31, 2011.

1 Introduction

We focus on discrete event systems with a structure of parallel processing, synchronization and non-concurrency. Typical examples of this kind of system include: manufacturing systems, transportation systems and project management, and so on. It is known that the behavior of this kind of system can be described using max-plus algebra [1][2], a class of Dioid algebra [3]. In this kind of systems, initial schedule is frequently changed due to unpredictable state change. The state change we mention here means a significant change in the schedule of tasks from the initial one. In general, giving buffers to a system as well as monitoring and controlling the tasks are effective for controlling such changes. Focusing on Max-Plus Linear (MPL) discrete event systems, there are several researches which consider uncertainties in the execution times of tasks [4]. We also examined a method based on Critical Chain Project Management (CCPM) method [5] for controlling such change [6].

The CCPM has been turn out to be an effective tool to protect projects from delays; it is an outgrowth of the theory of constraints (TOC), developed by Goldratt [7], for scheduling and management of manufacturing systems. In the CCPM, an empirical value is used to estimate the duration time of each process. Moreover, the CCPM provides a method for determining locations at which the time buffers should be inserted to prevent delays from the project's completion time.

Furthermore, the CCPM also provides a solution for problems of resource conflict. The resource conflict is a scramble for the resource which cannot be shared by multiple processes at the same time lines. If a resource conflict occurs, there is a risk that a delay occurs in the project. In general, the CCPM deals with both a single project and multiple projects cases. However, we consider a case of a single project, and assume that all processes are performed with the same resource.

In our previous paper [8], we modified a method for inserting time buffers in a previous research [6]. However, we have not considered the problem of resource conflict for simplicity. Therefore, we propose a method for resolving resource conflict in a single project in the MPL-CCPM representation.

This paper is organized as follows. In Section 2, we give an overview of max-plus algebra and MPL discrete event systems. In Section 3, we overview the concept of CCPM. In Section 4, we propose a method for resolving resource conflict for a single project in the MPL-CCPM representation. In Section 5, a simple model and numerical examples are presented. Finally, in Section 6, we summarize and conclude our work.

2 Max-Plus Linear System

We briefly review the max-plus algebra and MPL representation, both of which play essential roles throughout.

2.1 Max-plus algebra

Max-plus algebra is an algebraic system which is suitable for describing a certain class of discrete event systems. In a field $\boldsymbol{D} = \mathbb{R} \cup \{-\infty\}$, operators for addition and multiplication are defined as:

$$x \oplus y = \max(x, y)\ ,\ \ x \otimes y = x + y\ , \tag{1}$$

where $\mathbb{R}$ is the real field. The symbol $\otimes$ corresponds to multiplication in conventional algebra, and we often suppress this symbol when no confusion is likely to arise. For instance, we simply write xy as the simplified expression of $x \otimes y$. These operators hold the commutative, associative and distributive laws. The unit elements for these are denoted by $\varepsilon(=-\infty)$ and $e(=0)$, respectively. Moreover, the following two operators are defined for subsequent discussions:

$$x \odot y = -x + y\ ,\ \ x \wedge y = \min(x, y)\ . \tag{2}$$

An operator for the power of real numbers $a \in \mathbb{R}$ is defined as:

$$x^{\otimes a} = x \times a . \tag{3}$$

Operators for multiple numbers are as follows. If $m \le n$,

$$\bigoplus_{k=m}^{n} x_k = \max(x_m, x_{m+1}, \ldots, x_n) , \tag{4}$$

$$\bigwedge_{k=m}^{n} x_k = \min(x_m, x_{m+1}, \ldots, x_n) . \tag{5}$$

For matrices $\boldsymbol{X} \in \boldsymbol{D}^{m\times n}$, $[\boldsymbol{X}]_{ij}$ expresses the (i, j)-th element of $\boldsymbol{X}$, and $\boldsymbol{X}^T$ is the transpose matrix of $\boldsymbol{X}$. For $\boldsymbol{X}, \boldsymbol{Y} \in \boldsymbol{D}^{m\times n}$,

$$[\boldsymbol{X} \oplus \boldsymbol{Y}]_{ij} = \max\left([\boldsymbol{X}]_{ij} , [\boldsymbol{Y}]_{ij}\right), \tag{6}$$

$$[\boldsymbol{X} \wedge \boldsymbol{Y}]_{ij} = \min\left([\boldsymbol{X}]_{ij} , [\boldsymbol{Y}]_{ij}\right). \tag{7}$$

If $\boldsymbol{X} \in \boldsymbol{D}^{m\times l}$, $\boldsymbol{Y} \in \boldsymbol{D}^{l\times p}$

$$[\boldsymbol{X} \otimes \boldsymbol{Y}]_{ij} = \bigoplus_{k=1}^{l}\left([\boldsymbol{X}]_{ik} + [\boldsymbol{Y}]_{kj}\right), \tag{8}$$

$$[\boldsymbol{X} \odot \boldsymbol{Y}]_{ij} = \bigwedge_{k=1}^{l}\left(-[\boldsymbol{X}]_{ik} + [\boldsymbol{Y}]_{kj}\right), \tag{9}$$

where the priority of operators $\otimes$ and $\odot$ are higher than operators $\oplus$ and $\wedge$. Unit elements for matrices are: $\boldsymbol{\varepsilon}_{mn}$ is a matrix whose all elements are ε in $\boldsymbol{\varepsilon}_{mn} \in \boldsymbol{D}^{m\times n}$, and $\boldsymbol{e}_n$ is a matrix whose diagonal elements are e and off-diagonal elements are ε in $\boldsymbol{e}_n \in \boldsymbol{D}^{n\times n}$.

Operator % for matrix $\boldsymbol{X}$ is defined as:

$$[\boldsymbol{X}^{\%}]_{ij} = \begin{cases} e: & \text{if } [\boldsymbol{X}]_{ij} \text{ is negative,} \\ \varepsilon: & \text{if } [\boldsymbol{X}]_{ij} \text{ is } e \text{ or positive.} \end{cases} \tag{10}$$

2.2 Max-plus linear representation

We briefly review the process for deriving the max-plus linear representation for a certain class of discrete event systems developed in [8]. We assume that the following constraints are imposed on the focused system:

- The number of processes, external inputs and external outputs are n, p and q, respectively.
- All processes are used only once for a single job.
- Processing of the subsequent job cannot start while the process is at works for the current job.
- Processes with precedence constraints cannot begin processing until they will have received all required parts from the preceding processes.
- Processes with external inputs cannot begin until all required materials will have arrived.
- Processing starts as soon as all of the conditions above are satisfied.

For the k-th job in process $i (1 \le i \le n)$, let $d_i(k)(\ge 0)$, $[\boldsymbol{x}^-(k)]_i$, $[\boldsymbol{x}^+(k)]_i$, $[\boldsymbol{u}(k)]_i$ and $[\boldsymbol{y}(k)]_i$ be the processing, processing start, process completion, external input, and external output times, respectively. Moreover, matrices $\boldsymbol{P}_k$, $\boldsymbol{F}_0$, $\boldsymbol{B}_0$ and $\boldsymbol{C}_0$ given below are introduced for representing the structure of the system.

$$[\boldsymbol{P}_k]_{ij} = \begin{cases} d_i(k): & \text{if} \quad i = j, \\ \varepsilon: & \text{otherwise}, \end{cases} \tag{11}$$

$$[\boldsymbol{F}_0]_{ij} = \begin{cases} e: & \text{if process } i \text{ has a preceding process } j, \\ \varepsilon: & \text{if process } i \text{ does not have a preceding process } j, \end{cases} \tag{12}$$

$$[\boldsymbol{B}_0]_{ij} = \begin{cases} e: & \text{if process } i \text{ has an external input } j, \\ \varepsilon: & \text{if process } i \text{ does not have an external input } j, \end{cases} \tag{13}$$

$$[\boldsymbol{C}_0]_{ij} = \begin{cases} e: & \text{if process } j \text{ has an external output } i, \\ \varepsilon: & \text{if process } j \text{ does not have an external output } i, \end{cases} \tag{14}$$

where $\boldsymbol{F}_0$ is referred to as the adjacency matrix.

The earliest completion time is defined as the minimum value at which the corresponding process can complete processing. Then, the earliest completion times of all processes are given by [9]:

$$\boldsymbol{x}_E^+(k) = (\boldsymbol{P}_k\boldsymbol{F}_0)^*\boldsymbol{P}_k[\boldsymbol{x}^+(k-1) \oplus \boldsymbol{B}_0\boldsymbol{u}(k)], \tag{15}$$

where:

$$(\boldsymbol{P}_k\boldsymbol{F}_0)^* = \boldsymbol{e}_n \oplus \boldsymbol{P}_k\boldsymbol{F}_0 \oplus \cdots \oplus (\boldsymbol{P}_k\boldsymbol{F}_0)^{l-1}, \tag{16}$$

and an instance $l\,(1 \le l \le n)$ depends on the precedence-relationships of the system. The corresponding output times are given by:

$$\boldsymbol{y}_E(k) = \boldsymbol{C}_0\boldsymbol{x}_E^+(k). \tag{17}$$

Furthermore, the latest starting time is defined as the maximum value by which the same output time base on the earliest time can be accomplished. The latest starting times for all processes are given by [9]:

$$\boldsymbol{x}_L^-(k) = [(\boldsymbol{P}_k\boldsymbol{F}_0)^*\boldsymbol{P}_k]^T \odot [\boldsymbol{C}_0^T \odot \boldsymbol{y}_E(k)]. \tag{18}$$

Moreover, the latest input times are given by:

$$\boldsymbol{u}_L(k) = \boldsymbol{B}_0^T \odot \boldsymbol{x}_L^-(k). \tag{19}$$

Critical path is defined as the set of processes with zero total float. The total float is defined as the sum of float times in processes. This can also be obtained as the difference between the following two times; 1. the minimum value of the latest starting times in the succeeding processes, by which the output time is invariant, and 2. the completion time in the corresponding process included by the earliest starting time. Thus, the total floats in all processes are obtained as:

$$\boldsymbol{\omega}(k) = [\boldsymbol{x}_L^-(k)]_i - \left([\boldsymbol{x}_E^+(k)]_i - d_i(k)\right). \tag{20}$$

Then, the critical path is determined by the set of process numbers α that satisfy:

$$\text{Critical paths: } \{\alpha \,|\, [\boldsymbol{\omega}(k)]_\alpha = 0\}. \tag{21}$$

3 Critical Chain Project Management

Projects often exceed the initially planned duration. This is usually due to unforeseen uncertainties related to external factors. To resolve this problem, the CCPM is frequently useful [5]. The CCPM addresses several shortcomings of the Program Evaluation and Review Technique (PERT), the most widely used tool for project management. The PERT is based on identifying a critical path, which is the longest chain of the linked processes in the entire project. Focusing only on the longest chain of processes may result in several problems, such as resource conflict. On the other hand, the CCPM provides a solution for problems of resource conflict. Moreover, the CCPM provides a method for determining locations where time buffers should be inserted to prevent delay in the project. In the PERT, each process in the project consists a set of four times: the earliest start, earliest output, latest start, and latest output times. Since these times are observable by everyone involved in the project, they can be monitored closely. The difference between the earliest and the latest start times is equivalent to the slack time. Since the processes on the critical path do not have a slack time, significant attention should be paid for these processes. In estimating the process duration, we tend to use a safety estimate, which includes a significant margin to observe the due date. This value is often referred to as the 90% estimate. In the CCPM, an empirical value referred to as the ABP (Aggressive But Possible) time is used. The ABP is the time to complete the process with 50% probability. Specifically, we use $\mathrm{ABP} = \mathrm{HP}^{\otimes\frac{1}{3}}$ [5], where HP (Highly Possible) is the time to complete the process with 90% probability.

The next step is to resolve the resource conflict. A resource conflict occurs if the same resource is allocated for multiple processes at the same time lines. In the CCPM, problems of resource conflict are resolved based on the following procedures:

1. Detect resource conflicts.

2. Resolve the conflicts.

We can detect resource conflicts by confirming the allocation of resource and time line of each process. Moreover, in order to resolve the conflict for the case of a single project, a priority is attached for each process. If a resource conflict is detected between multiple processes, the processes with low priorities are moved up.

Moreover, the CCPM is supposed to insert a time buffer to absorb the uncertainty of task durations. This is referred to as the project buffer, which absorbs variations on the critical path. A project buffer is inserted between the final process on the critical path and the external output. Feeding buffers are inserted on the eve of processes on the critical chain that joins non-critical paths. The role of the feeding buffer is to protect the critical chain from variations of processing times on non-critical paths.

Finally, the critical path is monitored at which rate the project buffer is consumed. In this paper, we do not discuss about inserting time buffers as well as monitoring and controlling the buffers. These topics were discussed in [6], [8] and [10].

4 Proposed Method

We propose a method for resolving resource conflict for a single project in the MPL-CCPM representation. In addition, we assume that all processes are assigned on the same resource.

4.1 Detection of a resource conflict

We discuss how to detect a resource conflict between multiple processes. In order to detect this, we should examine whether the time lines of arbitrary two processes overlap. For process i, let $[\boldsymbol{x}^-]_i$ and $[\boldsymbol{x}^+]_i$ be the earliest starting and completion times, respectively. Similarly, for process j, let $[\boldsymbol{x}^-]_j$ and $[\boldsymbol{x}^+]_j$ be the earliest starting and completion times, respectively. In order to detect a resource conflict between processes i and j, we calculate the following index:

$$c = \max\{\left([\boldsymbol{x}^-]_i - [\boldsymbol{x}^+]_j\right), \left([\boldsymbol{x}^-]_j - [\boldsymbol{x}^+]_i\right), [\boldsymbol{e}_n]_{ij}\}. \tag{22}$$

If $c < 0$, a resource conflict occurs between processes i and j. If $i = j$, the same process is compared and $c < 0$ is followed. However, we do not regard this situation as a conflict. Thus, we add a term $[\boldsymbol{e}_n]_{ij}$ to satisfy $c \geq 0$.

In order to detect a resource conflict between multiple processes, we use the following theorem.

Theorem 1. *If* $[\boldsymbol{H}]_{ij} < 0$*, a resource conflict occurs between process* i *and* j.

$$\boldsymbol{H} = \boldsymbol{G} \oplus \boldsymbol{G}^T \oplus \boldsymbol{e}_n, \tag{23}$$

where:

$$\boldsymbol{G} = \boldsymbol{x}_E^+(k) \odot \boldsymbol{x}_E^-(k)^T, \tag{24}$$

$$\boldsymbol{x}_E^-(k) = \boldsymbol{P}_k \odot \boldsymbol{x}_E^+(k). \tag{25}$$

Proof. From Eq. (23):

$$[\boldsymbol{H}]_{ij} = [\boldsymbol{G}]_{ij} \oplus [\boldsymbol{G}]_{ji} \oplus [\boldsymbol{e}_n]_{ij}. \tag{26}$$

We call this matrix $\boldsymbol{H}$ as a conflict detection matrix. Moreover, from Eq. (24),

$$[\boldsymbol{G}]_{ij} = \left([\boldsymbol{x}_E^+(k)]_i \odot [\boldsymbol{x}_E^-(k)]_j\right) = [\boldsymbol{x}_E^-(k)]_j - [\boldsymbol{x}_E^+(k)]_i. \tag{27}$$

The element of matrix $[\boldsymbol{G}]_{ij}$ means the result of subtracting the earliest completion time from the earliest starting time of each process. Thus, Eq. (26) is represented as follows:

$$\begin{aligned}[\boldsymbol{H}]_{ij} &= [\boldsymbol{G}]_{ij} \oplus [\boldsymbol{G}]_{ji} \oplus [\boldsymbol{e}_n]_{ij} \\ &= \left([\boldsymbol{x}_E^-(k)]_i - [\boldsymbol{x}_E^+(k)]_j\right) \oplus \left([\boldsymbol{x}_E^-(k)]_j - [\boldsymbol{x}_E^+(k)]_i\right) \oplus [\boldsymbol{e}_n]_{ij} \\ &= \max\left\{\left([\boldsymbol{x}_E^-(k)]_i - [\boldsymbol{x}_E^+(k)]_j\right),\ \left([\boldsymbol{x}_E^-(k)]_j - [\boldsymbol{x}_E^+(k)]_i\right),\ [\boldsymbol{e}_n]_{ij}\right\}.\end{aligned} \tag{28}$$

We can now find that Eq. (28) is equivalent to Eq. (22). Therefore, if $[\boldsymbol{H}]_{ij} < 0$, we can detect that a resource conflict occurs between processes i and j. □

4.2 Resolution of a resource conflict

We discuss how to resolve a resource conflict between multiple processes. In this paper, we modify the precedence constraints of the original structure.

In order to resolve a resource conflict, priority is attached to each process. If a resource conflict is detected between multiple processes, by the policy in the CCPM method, the process with low priority is moved up. Therefore, in order to attach a priority to each process, we introduce the following priority vector $\boldsymbol{p}(\boldsymbol{p} \in \mathbb{N}^n)$. If the priority of process i is higher than process j,

$$[\boldsymbol{p}]_i < [\boldsymbol{p}]_j, \tag{29}$$

where $1 \le [\boldsymbol{p}]_i \le n (1 \le i \le n)$ holds. Moreover, in order to compare the priority of each process, we define the following matrix $\boldsymbol{M}$:

$$\boldsymbol{M} = \boldsymbol{p} \odot \boldsymbol{p}^T. \tag{30}$$

We refer to matrix $\boldsymbol{M}$ as the priority matrix. If $[\boldsymbol{M}]_{ij} < 0$, the priority of process i is lower than process j. If a resource conflict occurs between processes i and j, $[\boldsymbol{H}]_{ij}$ in Eq.(26) is negative. Similarly, if the priority of process i is lower than process j, $[\boldsymbol{M}]_{ij}$ is negative. Using matrices $\boldsymbol{H}$ and $\boldsymbol{M}$, we define a matrix $\boldsymbol{L}$ as follows:

$$\boldsymbol{L} = \boldsymbol{H} \oplus \boldsymbol{M}. \tag{31}$$

If $[\boldsymbol{L}]_{ij} < 0$, process i must be located before process j. This is because the resource conflict occurs between processes i and j. In addition, the priority of process i is lower than process j. Since the process i is moved up, we can confirm that the precedence constraints are changed from the original structure. Therefore, using Eq. (10), the modified adjacency matrix $\boldsymbol{F}_s$ after the resource conflicts have been resolved is defined as follows:

$$\boldsymbol{F}_s = \boldsymbol{F}_0 \oplus (\tilde{\boldsymbol{L}})^T , \tag{32}$$

where the subscript "s" expresses that the resource conflicts are resolved.

Using this modified adjacency matrix $\boldsymbol{F}_s$ and the framework in Section 2.2, we can reschedule the system with the resource conflicts being resolved.

5 Numerical Example

A simple model and numerical examples are presented to facilitate better understanding of the proposed method.

5.1 A simple system

We apply the method introduced in Section 2.2 for calculating the earliest and latest times and finding a critical path. Figure 1 shows a simple production system with one input, one output and four processes. The matrices defined in Eqs. (11) – (14) are given as:

$$\boldsymbol{P}_k = \mathrm{diag}(3,\ 15,\ 6,\ 3), \qquad \boldsymbol{F}_0 = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ e & \varepsilon & \varepsilon & \varepsilon \\ e & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & e & e & \varepsilon \end{bmatrix}, \tag{33}$$

$$\boldsymbol{B}_0 = \begin{bmatrix} e & \varepsilon & \varepsilon & \varepsilon \end{bmatrix}^T, \qquad \boldsymbol{C}_0 = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & e \end{bmatrix}. \tag{34}$$

Assuming that the initial condition is $\boldsymbol{x}(0) = \boldsymbol{\varepsilon}_{41}$ and the input time from the external input is $\boldsymbol{u} = [0]$, the earliest completion time $\boldsymbol{x}_E^+$ and the corresponding output time $\boldsymbol{y}_E$ are calculated using Eqs. (15) and (17) as follows:

$$\boldsymbol{x}_E^+ = \begin{pmatrix} 3 & 18 & 9 & 21 \end{pmatrix}^T, \qquad \boldsymbol{y}_E = 21. \tag{35}$$

From Eqs. (18) – (20), the latest starting times $\boldsymbol{x}_L^-$, input times $\boldsymbol{u}_L$, and the total floats $\boldsymbol{\omega}$ are obtained as:

$$\boldsymbol{x}_L^- = \begin{pmatrix} e & 3 & 12 & 18 \end{pmatrix}^T, \qquad \boldsymbol{u}_L = 0, \tag{36}$$

$$\boldsymbol{\omega} = \begin{pmatrix} e & e & 9 & e \end{pmatrix}^T. \tag{37}$$

Using these results, the critical path can be identified as $\alpha = \{1, 2, 4\}$ and we depict a Gantt chart of the system in Figure 2.

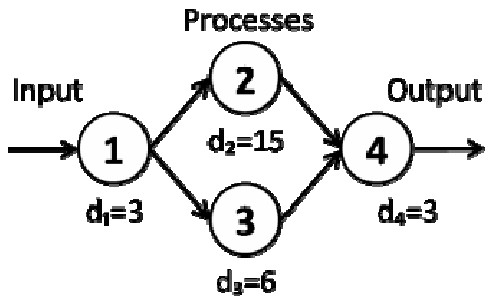

Figure 1: A simple production system

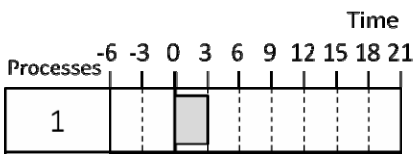

Figure 2: Gantt chart of the system of Figure 1

5.2 Resolution of the resource conflict

We resolve the resource conflict for the system shown in the previous subsection. First, we detect a resource conflict in Figure 1. Matrix $\boldsymbol{G}$ and the earliest starting times $\boldsymbol{x}_E^-$ are calculated using Eqs. (24) and (25) as follows:

$$\boldsymbol{G} = \begin{bmatrix} -3 & e & e & 15 \\ -18 & -15 & -15 & e \\ -9 & -6 & -6 & 9 \\ 21 & -18 & -18 & -3 \end{bmatrix}, \tag{38}$$

$$\boldsymbol{x}_E^- = \begin{pmatrix} e & 3 & 3 & 18 \end{pmatrix}^T. \tag{39}$$

Furthermore, from Eq. (23), the conflict detection matrix $\boldsymbol{H}$ is given as:

$$\boldsymbol{H} = \begin{bmatrix} e & e & e & 15 \\ e & e & -6 & e \\ e & -6 & e & 9 \\ 15 & e & 9 & e \end{bmatrix}. \tag{40}$$

Since $[\boldsymbol{H}]_{23} = [\boldsymbol{H}]_{32} < 0$, we can find that a resource conflict occurs between processes 2 and 3. In view of Figure 2, we can also confirm that the resource conflict occurs between processes 2 and 3. Next, we resolve the detected conflict. From Eq. (29), we set the priority vector $\boldsymbol{p}$ for each process as follows:

$$\boldsymbol{p} = \begin{pmatrix} 1 & 2 & 3 & 4 \end{pmatrix}^T. \tag{41}$$

Moreover, from Eq. (30), the priority matrix $\boldsymbol{M}$ is calculated as:

$$\boldsymbol{M} = \begin{bmatrix} e & 1 & 3 & 2 \\ -1 & e & 2 & 1 \\ -3 & -2 & e & -1 \\ -2 & -1 & 1 & e \end{bmatrix}. \tag{42}$$

Therefore, from Eq. (31), matrix $\boldsymbol{L}$ is given as:

$$\boldsymbol{L} = \begin{bmatrix} e & 1 & 3 & 15 \\ e & e & 2 & 1 \\ e & -2 & e & 9 \\ 15 & e & 9 & e \end{bmatrix}. \tag{43}$$

Since $[\boldsymbol{L}]_{32} < 0$, we can find process 3 must be moved up. This is because a resource conflict occurs between processes 2 and 3. In addition, the priority of process 3 is lower than process 2. Therefore, using Eq. (32), the modified adjacency matrix $\boldsymbol{F}_s$ after the resource conflicts have been resolved is obtained as:

$$\boldsymbol{F}_s = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ e & \varepsilon & e & \varepsilon \\ e & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & e & e & \varepsilon \end{bmatrix}. \tag{44}$$

Figure 3 shows the precedence relationships after the resource conflict has been resolved. The dotted allows shown in the Figure 3 shows the new precedence constraints.

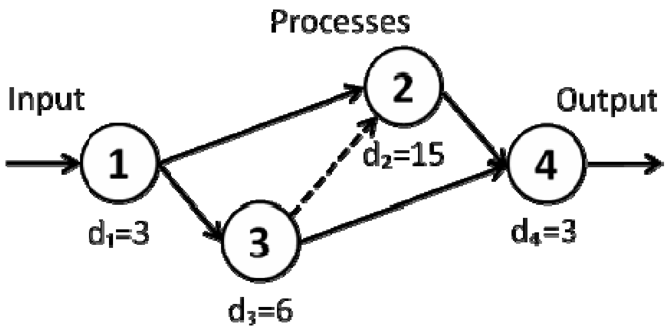

Figure 3: Precedence relationships after the resource conflict has been resolved

Using the modified adjacency matrix $\boldsymbol{F}_s$ in Eq. (44), we calculate the earliest and latest times and find a critical path again. Assuming that the output time from the external output is $\boldsymbol{y}_E = 21$ from Eq. (35), the latest starting time

$\boldsymbol{x}_{Ls}^{-}$ and input times $\boldsymbol{u}_{Ls}$ are calculated using Eqs. (18) and (19) as follows:

$$\boldsymbol{x}_{Ls}^{-} = \begin{pmatrix} -6 & 3 & -3 & 18 \end{pmatrix}^{T}, \quad \boldsymbol{u}_{Ls} = -6. \tag{45}$$

From Eqs. (15), (17) and (20), the earliest completion times $\boldsymbol{x}_{Es}^{+}$, output times $\boldsymbol{y}_{Es}$, and the total floats $\boldsymbol{\omega}_s$ are obtained as:

$$\boldsymbol{x}_{Es}^{+} = \begin{pmatrix} -3 & 18 & 3 & 21 \end{pmatrix}^{T}, \quad \boldsymbol{y}_{Es} = 21, \tag{46}$$

$$\boldsymbol{\omega}_s = \begin{pmatrix} e & e & e & e \end{pmatrix}^{T}. \tag{47}$$

Using these results, the critical path can be identified as $\alpha = \{1,2,3,4\}$.

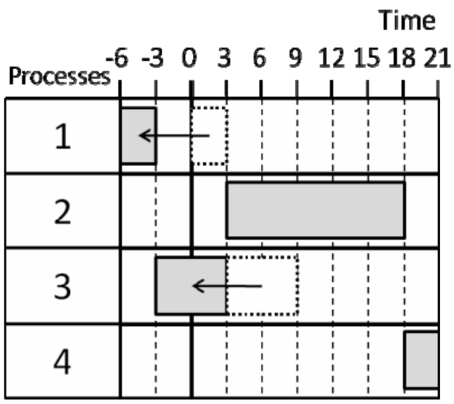

Figure 4: Gantt chart of the system of Figure 3

Consequently, we find that process 3 is moved up, because the precedence constraints are changed from the original structure. In addition, since processes 1 and 3 were moved up, the earliest completion, latest start, and input times for each process are moved up. Moreover, since the precedence constraints are changed from the original structure, process 3 is on the critical path. We depict the Gantt chart for the calculation results in Figure 4. In view of this chart, we can also find that the effect of the detection of process 3 is affected to process 1. Thus, the precedence constraints are changed.

In conclusion, using the modified adjacency matrix, we can confirm that the resource conflict in the single project is resolved.

6 Conclusion

We have proposed a method for resolving resource conflicts for a single project in the MPL-CCPM representation. To achieve this, the resource conflict was detected by subtracting completion time from starting time of each process. Moreover, in order to resolve the detected resource conflict, we attached a priority for each process. We detected processes that should be moved up. Since the precedence constraints were changed from the original structure, we defined new adjacency matrix after the resource conflict has been resolved. Therefore, using the new adjacency matrix, the resource conflict in the single project has been resolved.

In this paper, we assumed that all processes are allocated to the same resource for simplicity. Thus, in future work, we should develop a method which is applicable for cases where multiple resources are involved.

ACKNOWLEDGEMENTS. Hirotaka Takahashi has used the facilities of Earthquake Research Institute (ERI), The University of Tokyo and has been supported in part by JSPS Grant-in-Aid for Scientific Research No.23740207. Hiroyuki Goto has been supported in part by research grants from the Kayamori Foundation of Informational Science and the JSPS Grants-in-Aid for Scientific Research No. 20710114.

References

[1] G. Cohen, P. Moller, J. Quadrat and M. Viot, Algebraic tools for the performance evaluation of discrete event systems, *Proceedings of the IEEE*, **77**, (1989), 39 - 59.

[2] B. Heidergott, G. J. Olsder and L. Woude, *Max-plus at work*, Princeton University Press, New Jersey, 2006.

[3] F. Baccelli, G. Cohen, G. J. Olsder and J. P. Quadrat, *Synchronization and linearity, an algebra for discrete event systems*, John Wiley & Sons, New York, 1992.

[4] B. Heidergott, *Max Plus Linear Stochastic Systems and Perturbation Analysis*, Springer Verlag, New York, 2006.

[5] P. L. Lawrence, *Critical Chain Project Management*, Artech House Inc, 2005.

[6] H. Takahashi, H. Goto and M. Kasahara, Application of a critical chain project management based framework on max-plus linear systems, *International Journal of Computational Science*, **3**, (2009), 117 - 132.

[7] E. M. Goldratt, *Theory of constraints: and how it should be implemented*, North River Press, 1990.

[8] S. Yoshida, H. Takahashi and H. Goto, Modified max-plus linear representation for inserting time buffers, *Proceeding of The IEEE International Conference on Industrial Engineering Management,* (2010), 1631 - 1635.

[9] H. Goto, Dual representation of event-varying max-plus linear systems, *International Journal of Computational Science*, **1**, (2007), 225 - 242.

[10] M. Kasahara, H. Takahashi and H. Goto, On a buffer management policy for CCPM-MPL representation, *International Journal of Computational Science*, **3**, (2009), 593 - 606.