IEMS Vol. 10, No. 4, pp. 279-287, December 2011.

Resolution of Time and Worker Conflicts for a Single Project in a Max-Plus Linear Representation

Shotaro Yoshida†
Department of Management and Information Systems Engineering
Nagaoka University of Technology, Nagaoka, Niigata 940-2188, JAPAN
E-mail: s083381@stn.nagaokaut.ac.jp

Hirotaka Takahashi
Department of Humanities
Yamanashi Eiwa College, Kofu, Yamanashi 400-8555, JAPAN
E-mail: hirotaka@yamanashi-eiwa.ac.jp

Hiroyuki Goto
Department of Industrial and Management Systems
Hosei University, Koganei, Tokyo 184-8584, JAPAN
E-mail: goto-h@hosei.ac.jp

Received, September 28, 2011; Accepted, October 20, 2011

Abstract. This research develops a framework for resolving time and worker conflicts in the Critical Chain Project Management (CCPM) method, expressed in the form of a Max-Plus Linear (MPL) system. Our previous work proposed a method for resolving time conflicts. However, in practical cases, both time and worker conflicts may occur. Hence, we propose a method for resolving both time and worker conflicts for a single project. We first consider how to detect a resource conflict. Then, we define an adjacency matrix to resolve the detected conflicts. Using the proposed method, we confirm that the resource conflict can be resolved through a numerical example.

Keywords: Max-Plus Linear System, Critical Chain Project Management, Resource Conflict, Adjacency Matrix, Worker Conflict

1. INTRODUCTION

This research proposes a method for resolving resource conflicts in scheduling problems of a class of discrete event systems (DESs). We focus on the framework of resolving resource conflicts in the Critical Chain Project Management (CCPM) (Lawrence, 2005), which is represented in the form of Max-Plus Linear (MPL) system (Heidergott, 2006).

DESs with a structure of parallel processing, synchronization and non-concurrency typically appear in a class of manufacturing systems, transportation systems, project management, and so on. Max-plus algebra (Cohen *et al*., 1989; Heidergott *et al*., 2006; Baccelli *et al*., 1992) can describe the behavior of this kind of systems. A class of linear equations in max-plus algebra, called the MPL system, is known as an effective scheduling method for a class of DESs. However, it is difficult to take into account treat unpredictable delays that may influence on the project's completion time. This is because the MPL system plans an initial schedule using the processing times notified from workers in advance.

The CCPM has been turned out to be an effective tool to protect a project from delay; it is an outgrowth of the theory of constraints (TOC), developed by Goldratt (1990), for scheduling and managing manufacturing systems. The CCPM determine locations where time buffers should be inserted to avoid delay from the project's completion time.

Moreover, resource conflict is occasionally a reason of delay in the project. This is also a scramble for a resource which cannot be shared by multiple processes simultaneously. An occurrence of resource conflict may yield a risk of delay in a project; the CCPM provides a solution for this.

It is difficult to apply the concept of the CCPM to

† : Corresponding Author

large-scale systems because the precedence relations of processes are complicated, whereas MPL system can handle such systems easily. Thus, we recently developed an application framework of the CCPM to an MPL representation (Takahashi *et al.*, 2009a, 2009b). Specifically, we proposed a method for inserting time buffers, which are virtually handled as new processes. After the CCPM was applied, since the number of processes increased, the size of matrices for representing the structure of the system changed. Thus, it is difficult to understand the relationship between the original structure and the modified one. Therefore, we then proposed a method for inserting time buffers without installing new virtual processes (Yoshida *et al.*, 2010). Moreover, we proposed a method of monitoring a consumption of time buffers using a fever chart (Kasahara *et al.*, 2009), by which a project can be accomplished without delay. However, these frameworks do not consider an effect of resource conflict for simplicity. Therefore, Yoshida *et al.* (2011) proposed a method for resolving resource conflicts in a single project. In that paper, however, only a time conflict is considered for simplicity. However, in practical cases, both time and worker conflicts need to be considered.

Therefore, we propose a method for resolving resource conflicts with respect to time and worker for a single project. In this paper, we regard the overlaps of time-lines and workers as time and worker conflicts, respectively. If both of these conflicts occur at the same position, we regard this as a resource conflict. In this case, based on a policy of the CCPM, the process with low priority is moved up. To resolve the resource conflict, a new adjacency matrix is introduced.

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 the CCPM. In Section 4, we propose a method for resolving a 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. MATHEMATICAL BACKGROUND

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

2.1 Mathematical Preliminaries

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

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

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

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

An operator for the power of x is defined as:

$$x^{\otimes \alpha} = x \times \alpha, \tag{3}$$

where $\alpha \in \boldsymbol{R}$. Operators for multiple numbers are as follows. If $m \le n$,

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

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

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 $\boldsymbol{X}, \boldsymbol{Y} \in \boldsymbol{D}^{m\times n}$,

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

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

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

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

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

where the priority of operators $\otimes$ and $\odot$ are higher than operators $\oplus$ and $\wedge$. The zero and unit matrices are: $\boldsymbol{\varepsilon}$ is a matrix whose all elements are ε, and $\boldsymbol{e}$ is a matrix whose diagonal elements are e and off-diagonal elements are ε.

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

$$[\boldsymbol{X}^{\%}]_{ij} = \begin{cases} e: \ \text{if}[\boldsymbol{X}]_{ij} < e, \\ \varepsilon: \ \text{if}[\boldsymbol{X}]_{ij} \ge e. \end{cases} \tag{10}$$

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 Yoshida *et al.* (2010). 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 work 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.

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

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

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

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

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 Goto (2007):

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

where:

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

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 Goto (2007):

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

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. Moreover, 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. minimum value of the latest starting times in the succeeding processes, by which the output time is invariant, and 2. completion time in the corresponding process included by the earliest starting time. Thus, the total floats of all processes are obtained as:

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

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

$$\{\alpha \| [\boldsymbol{\varpi}(k)]_\alpha = 0\}. \tag{21}$$

3. CRITICAL CHAIN PROJECT MANAGEMENT

Projects often delay for initially planned schedule unpredicted uncertainties related to external factors. To resolve this problem, the CCPM is frequently useful (Lawrence, 2005). 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 avoid delay in the project. In the PERT, each process in the project is featured by the following 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 (Lawrence, 2005). 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 $\text{ABP} = \text{HP}^{\otimes 1/3}$ (Takahashi *et al.*, 2009b) 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 worker is allocated for multiple processes at the same time-lines. Although, the CCPM deals with both a single and multiple projects, we first focus on a method for resolving a resource conflict for a single project. In the CCPM, problems of resource conflict for a single project is resolved based on the following procedures: 1. Detect time conflicts, 2. Detect worker conflicts, and 3. Resolve the resource conflicts. If both the two conflicts occur at the same position simultaneously, we regard this as a resource conflict. Thus, we can detect resource conflicts by confirming the allocation of worker and time-line for each process. Moreover, in order to resolve the conflict for a single project, a priority is attached for each process. If a resource conflict is detected between multiple processes, 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 join 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, monitoring and controlling time buffers. These topics are discussed in Takahashi *et al.* (2009a, 2009b), Yoshida *et al.* (2010), Kasahara *et al.* (2009).

4. PROPOSED METHOD

We propose a method for resolving resource conflicts for a single project in the MPL-CCPM representation.

4.1 Detection of a Time Conflict

We discuss how to detect a time conflict between multiple processes. First, we examine whether the time-lines of arbitrary two processes overlap. For process i, let $[\boldsymbol{x}_E^-]_i$ and $[\boldsymbol{x}_E^+]_i$ be the earliest starting and com-pletion times, respectively. Similarly, for process j, let $[\boldsymbol{x}_E^-]_j$ and $[\boldsymbol{x}_E^+]_j$ be the earliest starting and completion times, respectively. If both $[\boldsymbol{x}_E^-]_i - [\boldsymbol{x}_E^+]_j$ and $[\boldsymbol{x}_E^-]_j - [\boldsymbol{x}_E^+]_i$ are negative, the time-lines of processes i and j overlap. Thus, in order to detect a time conflict between processes i and j, we calculate the following quantity:

$$[\boldsymbol{h}]_{ij} = \max\{([\boldsymbol{x}_{\boldsymbol{E}}^-]_i - [\boldsymbol{x}_{\boldsymbol{E}}^+]_j), ([\boldsymbol{x}_{\boldsymbol{E}}^-]_j - [\boldsymbol{x}_{\boldsymbol{E}}^+]_i)[\boldsymbol{e}]_{ij}\}. \quad (22)$$

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

In order to detect a time conflict between multiple processes, we use the following theorem:

Theorem 1: *Let matrix* $\boldsymbol{H}$ *be defined as*

$$\boldsymbol{H} = \boldsymbol{G} \oplus \boldsymbol{G}^T \oplus \boldsymbol{e}, \quad (23)$$

where

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

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

If $[\boldsymbol{H}]_{ij} < e$, *there is a time conflict between processes* i *and* j.

Proof: From Equation (23),

$$[\boldsymbol{H}]_{ij} = [\boldsymbol{G}]_{ij} \oplus [\boldsymbol{G}]_{ij} \oplus [\boldsymbol{e}]_{ij}. \quad (26)$$

Moreover, from Equation (24),

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

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

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

We can now find that Equation (28) is equivalent to Equation (22). Therefore, if $[\boldsymbol{H}]_{ij} < e$, there is a time conflict between processes i and j. □

4.2 Detection of a Worker Conflict

In order to detect a worker conflict, we first represent workers for each process. We introduce a vector $\boldsymbol{w}$ which represents the statuses of workers as follows:

$$[\boldsymbol{w}_i]_j = \begin{cases} e: \text{if worker } i \text{ is attached for process } j, \\ \varepsilon: \text{otherwise.} \end{cases} \quad (29)$$

where $1 \leq i \leq r (1 \leq j \leq n)$ holds, and r is the number of workers.

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

Theorem 2 : *Let matrix* $\boldsymbol{W}$ *be defined as*

$$\boldsymbol{W} = \bigoplus_{m=1}^{r} (\boldsymbol{w}_m \otimes \boldsymbol{w}_m^T). \quad (30)$$

If $[\boldsymbol{W}]_{ij} = e(i \neq j)$, *there is a worker conflict between processes i and j.*

Proof: From Equation (30),

$$\begin{aligned}[\boldsymbol{W}]_{ij} &= \bigoplus_{m=1}^{r}([\boldsymbol{w}_m]_i \otimes [\boldsymbol{w}_m]_j) \\ &= \max_{m=1,\cdots,r}([\boldsymbol{w}_m]_i + [\boldsymbol{w}_m]_j) \\ &= \max\{([\boldsymbol{w}_1]_i + [\boldsymbol{w}_1]_j), \cdots, ([\boldsymbol{w}_r]_i + [\boldsymbol{w}_r]_j)\}.\end{aligned} \tag{31}$$

If $([\boldsymbol{w}_m]_i + [\boldsymbol{w}_m]_j) = e$, for an instance, $[\boldsymbol{W}]_{ij} = e$, holds. Since $[\boldsymbol{w}_m]_i$ takes either of e or ε only, $([\boldsymbol{w}_m]_i + [\boldsymbol{w}_m]_j) = e$ is satisfied only if $[\boldsymbol{w}_m]_i = [\boldsymbol{w}_m]_j = e$. This means that the same worker is allocated to processes *i* and *j*. Note that if *i* = *j*, the same process is compared. However, we do not regard this situation as a conflict. In conclusion, if $[\boldsymbol{W}]_{ij} = e(i \neq j)$, we can detect a worker conflict between processes *i* and *j*. □

4.3 Resolution of a Resource Conflict

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

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

$$[\boldsymbol{p}]_i < [\boldsymbol{p}]_j, \tag{32}$$

holds where $1 \leq [\boldsymbol{p}]_i \leq n(1 \leq i \leq n)$. Moreover, to compare the priorities between all pair of processes, we use the following theorem.

Theorem 3: *Let matrix* $\boldsymbol{M}$ *be defined as*

$$\boldsymbol{M} = \boldsymbol{p} \odot \boldsymbol{p}^T. \tag{33}$$

If $[\boldsymbol{M}]_{ij} < e$, *the priority of process i is lower than that of process j.*

Proof: From Equation (33),

$$[\boldsymbol{M}]_{ij} = [\boldsymbol{p}]_j - [\boldsymbol{p}]_i. \tag{34}$$

If $[\boldsymbol{M}]_{ij} < e$, $[\boldsymbol{p}]_j < [\boldsymbol{p}]_i$ holds. This means that the priority of process *i* is lower than that of process *j*. □

In order to find a process that must be moved up, we use the following theorem.

Theorem 4: *Let matrix* $\boldsymbol{J}$ *be defined as*

$$\boldsymbol{J} = (\boldsymbol{L})^{\%} \wedge \boldsymbol{W}. \tag{35}$$

If $[\boldsymbol{J}]_{ij} = e$, *process i must be located on the eve of process j, where*

$$\boldsymbol{L} = \boldsymbol{H} \oplus \boldsymbol{M}. \tag{36}$$

Proof: The first term on the right hand-side of Equation (36) can be expanded as

$$\begin{aligned}[\boldsymbol{L}]_{ij} &= [\boldsymbol{H}]_{ij} \oplus [\boldsymbol{M}]_{ij} \\ &= \max\{([\boldsymbol{x}_E^-]_i - [\boldsymbol{x}_E^+]_j), ([\boldsymbol{x}_E^-]_j - [\boldsymbol{x}_E^+]_i), [\boldsymbol{e}]_{ij}\} \oplus ([\boldsymbol{p}]_j - [\boldsymbol{p}]_i) \\ &= \max\{([\boldsymbol{x}_E^-]_i - [\boldsymbol{x}_E^+]_j), ([\boldsymbol{x}_E^-]_j - [\boldsymbol{x}_E^+]_i), [\boldsymbol{e}]_{ij}, ([\boldsymbol{p}]_j - [\boldsymbol{p}]_i)\}.\end{aligned} \tag{37}$$

From theorems 1 and 3, if $[\boldsymbol{L}]_{ij} < e$, we can find that there is a time conflict between processes *i* and *j*. In addition, the priority of process *i* is lower than that of process *j*. Moreover, from Equation (35),

$$[\boldsymbol{J}]_{ij} = ([\boldsymbol{L}]_{ij})^{\%} \wedge [\boldsymbol{W}]_{ij}, \tag{38}$$

is followed. From Equation (10) and Equation (37), it is clear that $([\boldsymbol{L}]_{ij})^{\%}$ is e or ε. In addition, if $[\boldsymbol{L}]_{ij} < e$, $([\boldsymbol{L}]_{ij})^{\%} = e$ holds. On the other hand, since $\boldsymbol{W}$ is a logical matrix, $[\boldsymbol{W}]_{ij}$ is also e or ε. Thus, Equation (38) is represented as follows:

$$[\boldsymbol{J}]_{ij} = \begin{cases} e: \text{ if } ([\boldsymbol{L}]_{ij})^{\%} = [\boldsymbol{W}]_{ij} = e, \\ \varepsilon: \text{ otherwize.} \end{cases} \tag{39}$$

If there is a time conflict between processes *i* and *j*, and the priority of process *i* is lower than process *j*, $([\boldsymbol{L}]_{ij})^{\%} = e$ holds. In addition, if, there is a worker conflict between processes *i* and *j*, $[\boldsymbol{W}]_{ij} = e(i \neq j)$ holds. Therefore, if $[\boldsymbol{J}]_{ij} = e$, process *i* is must be located on the eve of process *j*. □

Since a process is moved up, we modify the precedence constraints from the original structure using the following theorem.

Theorem 5: *Let matrix* $\boldsymbol{F}_s$ *be defined as*

$$\boldsymbol{F}_s = \boldsymbol{F}_0 \oplus \boldsymbol{J}^T. \tag{40}$$

If $[\boldsymbol{F}_0]_{ij} = \varepsilon$ *and* $[\boldsymbol{J}]_{ij} = e$, *the modified adjacency matrix* $[\boldsymbol{F}_s]_{ij} = e$ *holds, and the new precedence*

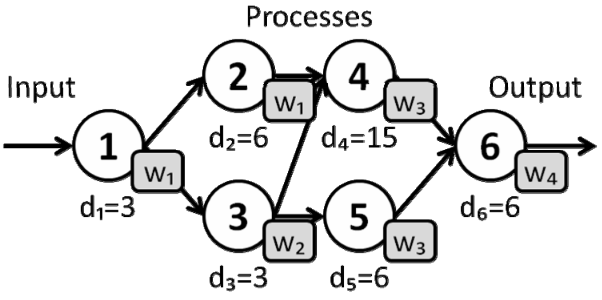

Figure 1. A Simple Production System.

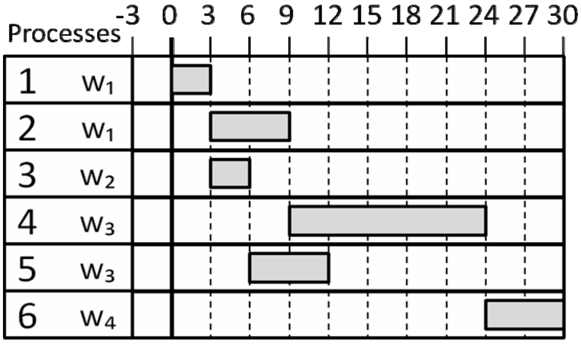

Figure 2. Gantt Chart of the system of Figure 1.

constraints from process j to i is added. The subscript 's' expresses that the resource conflicts in a single project are resolved already.

Proof: From Equation (40),

$$\begin{aligned}[\boldsymbol{F}_s]_{ij} &= [\boldsymbol{F}_0]_{ij} \oplus [\boldsymbol{J}^T]_{ij} \\ &= [\boldsymbol{F}_0]_{ij} \oplus [\boldsymbol{J}]_{ji} = \max\{[\boldsymbol{F}_0]_{ij}, [\boldsymbol{J}]_{ji}\}.\end{aligned} \tag{41}$$

From this result, Equation (41) is represented as follows:

$$[\boldsymbol{F}_s]_{ij} = \begin{cases} e : \text{if}\, [\boldsymbol{F}_0]_{ij} = [\boldsymbol{J}]_{ji} = e, \\ e : \text{if}\, [\boldsymbol{F}_0]_{ij} = e, [\boldsymbol{J}]_{ji} = \varepsilon, \\ e : \text{if}\, [\boldsymbol{F}_0]_{ij} = \varepsilon, [\boldsymbol{J}]_{ji} = e, \\ e : \text{if}\, [\boldsymbol{F}_0]_{ij} = [\boldsymbol{J}]_{ji} = \varepsilon. \end{cases} \tag{42}$$

From Equation (12), if $[\boldsymbol{F}_0]_{ij} = e$, there is precedence constraint from the process *j* to *i*. If there is precedence constraint between processes *i* and *j*, there is not a time conflict between these processes. If $[\boldsymbol{F}_0]_{ij} = e$, there is not a resource conflict between processes *i* and *j*. Therefore, it is clear that $[\boldsymbol{F}_0]_{ij} = [\boldsymbol{J}]_{ji} = e$ does not hold. Moreover, if $[\boldsymbol{J}]_{ji} = \varepsilon, [\boldsymbol{F}_0]_{ij}$ is equal to $[\boldsymbol{F}_s]_{ij}$ which means that, the precedence constraint of original structure is preserved. Therefore, only if $[\boldsymbol{F}_0]_{ij} = \varepsilon$ and $[\boldsymbol{J}]_{ji} = e$, the precedence constraint is changed from the original structure. From theorem 4, if $[\boldsymbol{J}]_{ji} = e$, process *j* must be located on the eve of process *i*. Therefore, a precedence constraint from the process *j* to *i* is added. From Equation (12), if $[\boldsymbol{F}_s]_{ij} = e$, process *i* has a preceding process *j*. That is, there is a precedence constraint from process *j* to *i*. Consequently, if $[\boldsymbol{F}_0]_{ij} = \varepsilon$ and $[\boldsymbol{J}]_{ji} = e$, the modified adjacency matrix holds $[\boldsymbol{F}_s]_{ij} = e$, and the new precedence constraint from process *j* to *i* is added. □

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

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 six processes. The matrices defined in Equation (11)~Equation (14) are given as:

$$\boldsymbol{P}_k = \text{diag}(3, 6, 3, 15, 6, 6), \boldsymbol{F}_0 = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & e & e & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & e & e & \varepsilon \end{bmatrix}, \tag{43}$$

$$\boldsymbol{B}_0 = [e \;\; \varepsilon \;\; \varepsilon \;\; \varepsilon \;\; \varepsilon \;\; \varepsilon]^T, \boldsymbol{C}_0 = [\varepsilon \;\; \varepsilon \;\; \varepsilon \;\; \varepsilon \;\; \varepsilon \;\; e]. \tag{44}$$

Assuming that the initial condition is $\boldsymbol{x}(0) = \boldsymbol{\varepsilon} \in \boldsymbol{D}^{6\times 1}$ 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 Equation (15) and Equation (17) as follows:

$$\boldsymbol{x}_E^+ = (3 \;\; 9 \;\; 6 \;\; 24 \;\; 12 \;\; 30)^T, \; \boldsymbol{y}_E = 30. \tag{45}$$

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

$$\boldsymbol{x}_E^- = (e \;\; 3 \;\; 6 \;\; 9 \;\; 18 \;\; 24)^T, \boldsymbol{u}_L = 0, \tag{46}$$

$$\boldsymbol{\omega} = (e \;\; e \;\; 3 \;\; e \;\; 12 \;\; e)^T. \tag{47}$$

Using these results, the critical path is identified as $\alpha = \{1, 2, 4, 6\}$ and we depict a Gantt chart of the system in Figure 2.

5.2 Resolution of the Resource Conflict

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

$$\boldsymbol{x}_E^- = (e \;\; 3 \;\; 3 \;\; 9 \;\; 6 \;\; 24)^T, \tag{48}$$

$$\boldsymbol{G} = \begin{bmatrix} -3 & e & e & 6 & 3 & 21 \\ -9 & -6 & -6 & e & -3 & 15 \\ -6 & -3 & -3 & 3 & e & 18 \\ -24 & -21 & -21 & -15 & -18 & e \\ -12 & -9 & -9 & -3 & -6 & -12 \\ -30 & -27 & -27 & -21 & -24 & -6 \end{bmatrix}. \tag{49}$$

Furthermore, from Equation (23), matrix $\boldsymbol{H}$ is given as:

$$\boldsymbol{H} = \begin{bmatrix} e & e & e & 6 & 3 & 21 \\ e & e & -3 & e & -3 & 15 \\ e & -3 & e & 3 & e & 18 \\ 6 & e & 3 & e & -3 & e \\ 3 & -3 & e & -3 & e & 12 \\ 21 & 15 & 18 & e & 12 & e \end{bmatrix}. \tag{50}$$

Since $[\boldsymbol{H}]_{23} = [\boldsymbol{H}]_{32} < e$, $[\boldsymbol{H}]_{25} = [\boldsymbol{H}]_{52} < e$ and $[\boldsymbol{H}]_{45} = [\boldsymbol{H}]_{54} < e$, we can find that there are time conflicts between processes (2-3), (2-5), and (4-5). We can also confirm these time conflicts from Figure 2, which shows the Gantt chart of the system of Figure 1. Next, we detect a worker conflict in Figure 1. From Equation (29), we set the worker vector $\boldsymbol{w}$ for each process as follows:

$$\begin{aligned} \boldsymbol{w}_1 &= (e\ e\ \varepsilon\ \varepsilon\ \varepsilon\ \varepsilon)^T, \\ \boldsymbol{w}_2 &= (\varepsilon\ \varepsilon\ e\ \varepsilon\ \varepsilon\ \varepsilon)^T, \\ \boldsymbol{w}_3 &= (\varepsilon\ \varepsilon\ \varepsilon\ e\ e\ \varepsilon)^T, \\ \boldsymbol{w}_4 &= (\varepsilon\ \varepsilon\ \varepsilon\ \varepsilon\ \varepsilon\ e)^T. \end{aligned} \tag{51}$$

Moreover, from Equation (30), matrix $\boldsymbol{W}$ is calculated as:

$$\boldsymbol{W} = \begin{bmatrix} e & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ e & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & e & e & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & e & e & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e \end{bmatrix}. \tag{52}$$

Since $[\boldsymbol{W}]_{12} = [\boldsymbol{W}]_{21} = e$ and $[\boldsymbol{W}]_{45} = [\boldsymbol{W}]_{54} = e$, we can find there are worker conflicts between processes (1-2) and (4-5). We can also confirm these worker conflicts from Figure 2. Finally, we resolve the detected conflict. From Equation (32), we assume the priority vector $\boldsymbol{p}$ for each process as follows:

$$\boldsymbol{p} = (1\ 4\ 6\ 2\ 3\ 5)^T. \tag{53}$$

Moreover, from Equation (33), matrix $\boldsymbol{M}$ is calculated as:

$$\boldsymbol{M} = \begin{bmatrix} e & 3 & 5 & 1 & 2 & 4 \\ -3 & e & 2 & -2 & -1 & 1 \\ -5 & -2 & e & -4 & -3 & -1 \\ -1 & 2 & 4 & e & 1 & 3 \\ -2 & 1 & 3 & -1 & e & 2 \\ -4 & -1 & 1 & -3 & -2 & e \end{bmatrix}. \tag{54}$$

Thus, from Equation (36), matrix $\boldsymbol{L}$ is given as:

$$\boldsymbol{L} = \begin{bmatrix} e & 3 & 5 & 6 & 3 & 21 \\ e & e & 2 & e & -1 & 15 \\ e & -2 & e & 3 & e & 18 \\ 6 & 2 & 4 & e & 1 & 3 \\ 3 & 1 & 3 & -1 & e & 12 \\ 21 & 15 & 18 & e & 12 & e \end{bmatrix}. \tag{55}$$

Since $[\boldsymbol{L}]_{25} < e, [\boldsymbol{L}]_{32} < e$ and $[\boldsymbol{L}]_{54} < e$, we can find there are time conflicts between processes (2-5), (2-3) and (4-5). In addition, the priority of process 2 is lower than that of process 5. Similarly, the priority of process 3 is lower than that of process 2, and the priority of process 5 is lower than that of process 4. From Equation (35), using Equation (10), matrix $\boldsymbol{J}$ is given as:

$$\boldsymbol{J} = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{bmatrix}. \tag{56}$$

Since $[\boldsymbol{J}]_{54} = e$, we can find process 5 must be moved up. This is because time and worker conflicts occur between processes 4 and 5. In addition, the priority of process 5 is lower than that of process 4. Therefore, the modified adjacency matrix $\boldsymbol{F}_s$ after the resource conflicts have been resolved is obtained as:

$$\boldsymbol{W} = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & e & e & \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & e & e & \varepsilon \end{bmatrix}. \tag{57}$$

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

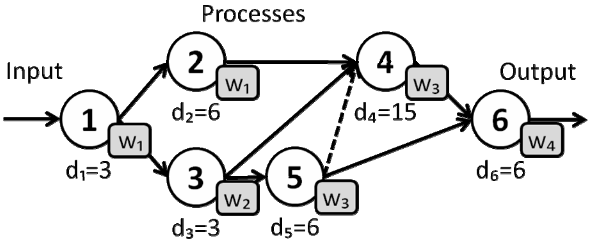

Figure 3. Precedence relationships after the resource conflict has been resolved.

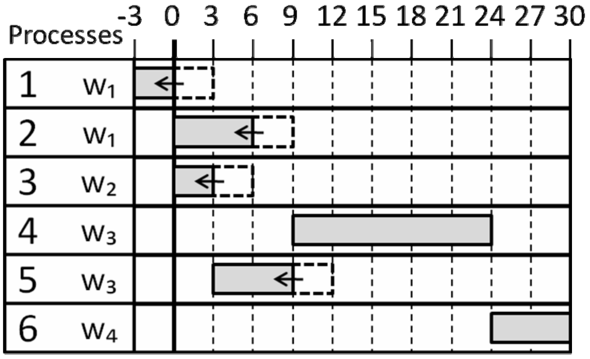

Figure 4. Gantt chart of the system of Figure 3.

Using the modified adjacency matrix $\boldsymbol{F}_s$ in Equation (57), we calculate the earliest and latest times and find a critical path again. Assuming that the output time from the external output is $\boldsymbol{y}_E = 30$ from Equation (45), the latest starting time $\boldsymbol{x}_{Ls}^{-}$ and input times $\boldsymbol{u}_{Ls}$ are calculated using Equation (18) and Equation (19) as follows:

$$\boldsymbol{x}_{Ls}^{-} = (-3\ 3\ e\ 9\ 3\ 24)^T, \boldsymbol{u}_{Ls} = -3. \quad (58)$$

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

$$\boldsymbol{x}_{Es}^{+} = (e\ 6\ 3\ 24\ 9\ 30)^T,\ \boldsymbol{y}_{Es} = 30, \quad (59)$$

$$\boldsymbol{\omega}_s = (e\ 3\ e\ e\ e\ e)^T. \quad (60)$$

Using these results, the critical path can be identified as $\alpha = \{1, 3, 4, 5, 6\}$.

Consequently, we find that process 5 is moved up, because the precedence constraints are changed from the original structure. In addition, since processes 1, 2, 3 and 5 are moved up, the earliest completion, latest start, and input times of the corresponding processes are moved up. Thus, the critical path has changed from the original one.

We detect time conflicts between processes (2-3), (2-5). However, since a worker conflict between the above processes is not detected, we can confirm that there is not a resource conflict between these processes. Similarly, we detect a worker conflict between processes 1 and 2. However, since a time conflict between these processes is not detected, we can confirm that there is not a resource conflict.

We depict a Gantt chart for the calculation results in Figure 4. In view of the chart, we can also find that the effect of the resource conflict in process 5 affects processes 1, 2 and 3. 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 time and worker conflicts for a single project in the MPL-CCPM representation. In the first step, time conflicts were detected by subtracting the completion time from the starting time for all processes. Then, worker conflicts were detected by assigning a worker for each process. In order to resolve a resource conflict, we attached a priority for each process. After that, we identified processes which should be moved up. Since the precedence constraints were changed from the original structure, we defined a new adjacency matrix after the resource conflicts were resolved. Using the new adjacency matrix, the resource conflicts were resolved.

Since we have considered a resolution of resource conflicts only for a single project, developing a method for multiple projects remains as our future work.

ACKNOWLEDGMENT

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 a research grant from JSPS Grants-in-Aid for Scientific Research No. 23510163.

REFERENCES

Baccelli, F., Cohen, G., Olsder, G. J., and Quadrat, J. P. (1992), *Synchronization and Linearity*, *An Algebra for Discrete Event Systems*, John Wiley and Sons, New York.

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

Goldratt, E. M. (1990), *Theory of Constraints: And How It Should Be Implemented*, North River Pr.

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

Heidergott, B., Olsder, G. J., and Woude, L. (2006), *Max Plus at Work: Modeling and Analysis of Synchronized Systems*, New Jersey, Princeton University Press.

Heidergott, B. (2006), *Max Plus Linear Stochastic Systems and Perturbation Analysis*, New York, Springer Verlag.

Kasahara, M., Takahashi, H., and Goto, H. (2009), On a buffer management policy for CCPM-MPL representation, *International Journal of Computational*

Science, **3**, 593-606.

Lawrence, P. L. (2005), *Critical Chain Project Management*, Artech House Inc.

Takahashi, H., Goto, H., and Kasahara, M. (2009a), Toward the application of a critical chain project management based framework on max-plus linear systems, *International Engineering and Management Systems*, **8**, 155-161.

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

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

Yoshida, S., Takahashi, H., Goto, H. (2011), Resolution of resource conflict for a single project in max-plus linear representation, *Journal of Computations and Modeling*, **1**, 31-47.