Toward the Resolution of Resource Conflict in a MPL-CCPM Representation Approach

MUNENORI KASAHARA
Nagaoka University of Technology
Nagaoka, Niigata 940-2188
JAPAN
s063349@ics.nagaokaut.ac.jp

HIROTAKA TAKAHASHI
Nagaoka University of Technology
Nagaoka, Niigata 940-2188
JAPAN
hirotaka@oberon.nagaokaut.ac.jp

HIROYUKI GOTO
Nagaoka University of Technology
Nagaoka, Niigata 940-2188
JAPAN
hgoto@kjs.nagaokaut.ac.jp

Abstract: Max-Plus-Linear (MPL) system is known as a solution for scheduling problems of a class of discrete event systems. Typical systems include: manufacturing system and project management, etc. In such systems, the initial schedule is frequently changed due to unpredictable disturbances. Critical Chain Project Management (CCPM) is a management method that can shorten the time required to complete a project based on the Theory Of Constraints (TOC). Therefore, in our previous papers [1], we proposed a method for applying the concepts of the CCPM to MPL representation in order to control the occurrence of an undesirable state change. The CCPM method also provides a solution for the problem of the resource conflict. However, we did not discuss the resource conflict. In this paper, we propose a method for discovering the resource conflict between two different projects in the MPL-CCPM representation.

Key–Words: Max-Plus-Linear system, Critical Chain Project Management, Resource Conflict, State-Space Representation, Scheduling.

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, etc. It is known that the behavior of this kind of system can be described using max-plus algebra [2], a subclass of Dioid algebra [3]. In this kind of systems, the initial schedule is frequently changed due to unpredictable disturbances.

In this context, we examine a method of controlling the occurrence of an undesirable state change in those systems. The state change we mention here means a significant change of the schedule of tasks from the initial one. In general, not only giving buffers to the system but monitoring and controlling the tasks in a wider range are effective for controlling such changes.

Focusing on Max Plus Linear (MPL) discrete event systems, there are several researches which consider uncertainty of the execution time of tasks [4]. However, if the relevant parameter contains stochastic variations, there is a strong non-linearity in the states of the system. Thus, it is difficult to handle large-scale systems.

On the other hand, the method based on CCPM method (e.g. [5] and the reference therein) has an advantage in that it can handle such systems easily. The CCPM method has been found to be an effective tool to protect projects from delays. The CCPM method is an outgrowth of the theory of constraints (TOC), developed by Goldratt [6] in 1990, for scheduling and management of manufacturing. In the CCPM method, an empirical value is used to obtain an estimate of the process duration for each process. Moreover, the CCPM method provides a method for determining places locations at which time buffers should be inserted in order to prevent unplanned delays in completing the project. This is because the method does not consider the change of execution points of individual tasks, and a buffer is incorporated into the cluster of tasks. Therefore, in our previous papers [1], we proposed a method for applying the concepts of the feeding buffer and the project buffer in MPL-CCPM discrete event systems. Moreover, we proposed a method of monitoring the buffer consumption with fever chart [5] to accomplish projects, which is a buffer management policy for CCPM-MPL representation [7].

The CCPM method also provides a solution for the problem of the resource conflict. However, in our previous papers [1] [7], we did not consider this problem for simplicity. Therefore, in this paper, we propose a method for discovering the resource conflict between two different projects in the MPL-CCPM representation.

This paper is organised as follows. In Sec.2, we give an overview of the Max-Plus algebra and the MPL discrete event systems. In Sec.3, we overview the concept of the CCPM. In Sec.4, we describe the methods for applying the concepts of the feeding and the project buffers in the MPL-CCPM discrete event systems. In Sec.5, we propose a method for discovering the resource conflict between two different projects in the MPL-CCPM representation. In Sec.6, a simple model and numerical examples are presented. Finally, in Sec.7, we summarize and conclude our work.

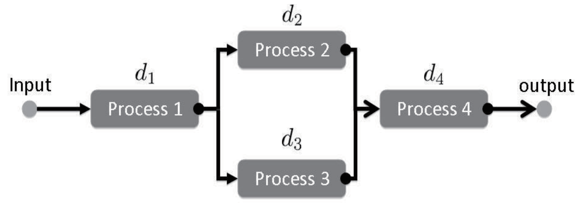

Figure 1: The manufacturing sequence of a simple production system.

2 Max-Plus Linear System

In this section, we briefly review the Max-Plus Algebra and the MPL discrete event systems, both of which would be the essential backgrounds throughout this paper.

2.1 Max-Plus Algebra

Max-Plus algebra is an algebraic system that is suitable for describing a certain class of discrete event systems. In a field $\mathcal{D} = \mathbf{R} \cup \{-\infty\}$, where $\mathbf{R}$ is the real field, the operators for addition and multiplication are defined as: $x \oplus y = \max(x, y), \quad x \otimes y = x + y$. The symbol $\otimes$ corresponds to the multiplication in conventional algebra, and we often suppress this notification when no confusion is likely to arise. For instance, we simply write xy as the simplified expression of $x \otimes y$. These hold the commutative, associative and distributive laws. By definition, the unit elements for these operators are given by $\varepsilon(\equiv -\infty)$ and $e(\equiv 0)$, respectively. The following relationships are satisfied for an arbitrary $x \in \mathcal{D}$: $\varepsilon \oplus x = x \oplus \varepsilon = x, \quad e \otimes x = x \otimes e = x$. Further, the following two operators are defined for subsequent discussions. $x \wedge y = \min(x, y), \quad x \setminus y = -x + y$. Denote the unit element of $\wedge$ by $\bar{\varepsilon}(\equiv +\infty)$. Moreover, we adopt the following axiom : $\varepsilon \otimes \bar{\varepsilon} = \bar{\varepsilon} \otimes \varepsilon = \varepsilon$. An operator for the powers of real numbers α is introduced as: $x^{\otimes\alpha} = \alpha \times x$ for $\alpha \in \mathbf{R}$. For matrices $\boldsymbol{X} \in \mathcal{D}^{m\times n}$, $[\boldsymbol{X}]_{ij}$ express the (i, j)-th element of $\boldsymbol{X}$, and $\boldsymbol{X}^T$ is the transpose matrix of $\boldsymbol{X}$. For $\boldsymbol{X}, \boldsymbol{Y} \in \mathcal{D}^{m\times n}$,

$$[\boldsymbol{X} \oplus \boldsymbol{Y}]_{ij} = \max([\boldsymbol{X}]_{ij}, [\boldsymbol{Y}]_{ij}), \quad (1)$$

$$[\boldsymbol{X} \wedge \boldsymbol{Y}]_{ij} = \min([\boldsymbol{X}]_{ij}, [\boldsymbol{Y}]_{ij}). \quad (2)$$

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

$$[\boldsymbol{X} \otimes \boldsymbol{Y}]_{ij} = \max_{k=1,\cdots,l}([\boldsymbol{X}]_{ik} + [\boldsymbol{Y}]_{kj}), \quad (3)$$

$$[\boldsymbol{X} \odot \boldsymbol{Y}]_{ij} = \min_{k=1,\cdots,l}(-[\boldsymbol{X}]_{ik} + [\boldsymbol{Y}]_{kj}), \quad (4)$$

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

2.2 Max-Plus Linear System and Representation

The Max Plus Linear (MPL) discrete event system is defined as a system whose behaviour can be described in linear form in Max-Plus Algebra, it is similar to the state-space equations in modern control theory.

$$\boldsymbol{x}(k) = \boldsymbol{A}(k)\boldsymbol{x}(k-1) \oplus \boldsymbol{B}\boldsymbol{u}(k), \quad (5)$$

$$\boldsymbol{y}(k) = \boldsymbol{C}\boldsymbol{x}(k), \quad (6)$$

where k is called the event counter that represents the number of event occurrences from the initial state. $\boldsymbol{x}(k) \in \mathcal{D}^n, \boldsymbol{u}(k) \in \mathcal{D}^p$ and $\boldsymbol{y}(k) \in \mathcal{D}^q$ are the states, input and output variables, respectively. n, p and q are the corresponding dimensions. $\boldsymbol{A} \in \mathcal{D}^{m\times n}$,$\boldsymbol{B} \in \mathcal{D}^{q\times n}$ and $\boldsymbol{C} \in \mathcal{D}^{p\times n}$ are called the system input and output matrices respectively. Let us consider a simple production system shown in Fig.1 as an example. The processing times in sequences No.1-4 are denoted as d_1, d_2, d_3 and d_4, respectively. For the k-th job, suppose $u(k), \boldsymbol{x}(k)$ and $y(k)$ represent the feeding, processing starting and finishing times, respectively. We then find the following relations.

$$x_1(k) = \max\{u(k), x_1(k-1) + d_1\}, \quad (7)$$

$$x_2(k) = \max\{x_2(k-1) + d_2, x_1(k) + d_1\}, \quad (8)$$

$$x_3(k) = \max\{x_3(k-1) + d_3, x_1(k) + d_1\}, \quad (9)$$

$$x_4(k) = \max\{x_4(k-1) + d_4, x_2(k) + d_2, x_3(k) + d_3\}, \quad (10)$$

$$y(k) = x_4(k) + d_4. \quad (11)$$

ISSN: 1790-2769

ISBN: 978-960-474-168-7

It can be found that the Eqs.(7)-(11) can be described in the forms of Eqs.(5) and (6) , where

$$\boldsymbol{A}=\begin{pmatrix} d_1 & \varepsilon & \varepsilon & \varepsilon \\ d_1^2 & d_2 & \varepsilon & \varepsilon \\ d_1^2 & \varepsilon & d_3 & \varepsilon \\ a_{41} & d_2^2 & d_3^2 & d_4 \end{pmatrix}, \boldsymbol{B}=\begin{pmatrix} e \\ d_1 \\ d_1 \\ b_4 \end{pmatrix}, \boldsymbol{C}=\begin{pmatrix} \varepsilon \\ \varepsilon \\ \varepsilon \\ d_4 \end{pmatrix}^T, \quad (12)$$

$a_{41} = d_1^2(d_2 \oplus d_3)$ and $b_4 = d_1(d_2 \oplus d_3)$.

Since there is a term $x_i(k)$ on the right hand side of Eqs.(8)-(10), they must be transformed into equations without the term $x_i(k)$. This implies that they are expressed in the form of Eq.(5). Thus, there has been few discussions about the domain, where equations are described as a general form of the system matrix and input/output matrices. Therefore, we briefly review a general formulation of the MPL equations for systems with precedence constrains or synchronizations that were developed in [8]. We assume that the relevant constraints are imposed on the focused system in the following way:

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

For the k-th job in process $i(1 \leq i \leq n)$, let $d_i(k)(\geq 0)$, $[\boldsymbol{x}(k)]_i$, $[\boldsymbol{u}(k)]_i$ and $[\boldsymbol{y}(k)]_i$ be the processing time, the starting time, external input time and external output time for each process. Matrices $\boldsymbol{A}_k^0, \boldsymbol{F}_k, \boldsymbol{B}^0$ and $\boldsymbol{C}_k^0$ are introduced for representing the structure of systems as follows.

$$\left[\boldsymbol{A}_k^0\right]_{ij}=\begin{cases} d_i(k) : \text{if } i=j. \\ \varepsilon \quad : \text{otherwise.} \end{cases} \quad (13)$$

$$[\boldsymbol{F}_k]_{ij}=\begin{cases} d_j(k) : \text{if process } i \text{ has a preceding process } j. \\ \varepsilon \quad : \text{if process } i \text{ does not have a preceding process } j. \end{cases} \quad (14)$$

$$\left[\boldsymbol{B}^0\right]_{ij}=\begin{cases} e \ : \text{if process } i \text{ has an external input } j. \\ \varepsilon \ : \text{if process } i \text{ dose not have an external input } j. \end{cases} \quad (15)$$

$$\left[\boldsymbol{C}_k^0\right]_{ij}=\begin{cases} d_j(k) : \text{if process } j \text{ has an external output } i. \\ \varepsilon \quad : \text{if process } j \text{ dose not have an external output } i. \end{cases} \quad (16)$$

$\boldsymbol{F}_k$ is referred to the adjacency matrix.

For the k-th job in process $i(1 \leq i \leq n)$, $[\boldsymbol{A}_k^0\boldsymbol{x}(k-1)]_i$ gives the finishing time, $[\boldsymbol{F}_k\boldsymbol{x}(k)]_i$ states the latest time among finishing times in the preceding processes, and $[\boldsymbol{B}^0\boldsymbol{u}(k)]_i$ is equal to the latest feeding time from external inputs. Furthermore, $[\boldsymbol{C}_k^0\boldsymbol{x}(k)]_i$ represents the latest time among the finishing times in processes attached to the corresponding output.

The earliest starting time is defined as the minimum value on which the corresponding process can start processing immediately. Utilising the above discussions, the earliest starting times of any of the processes are given by [8] :

$$\boldsymbol{x}_E(k) = \boldsymbol{F}_k^*\boldsymbol{A}_{k-1}^0\boldsymbol{x}(k-1) \oplus \boldsymbol{F}_k^*\boldsymbol{B}^0\boldsymbol{u}(k), \quad (17)$$

where

$$\boldsymbol{F}_k^*=\boldsymbol{e}_n \oplus \boldsymbol{F}_k \oplus \boldsymbol{F}_k^2 \oplus \cdots \oplus \boldsymbol{F}_k^{l-1},\ \boldsymbol{F}_k^l=\boldsymbol{\varepsilon}_{nn}. \quad (18)$$

An instance $l(1 \leq l \leq n)$ depends on the precedence-relations of the systems. The corresponding output times are given by:

$$\boldsymbol{y}_E(k) = \boldsymbol{C}_k^0\boldsymbol{x}(k). \quad (19)$$

Furthermore, the latest starting time is defined as the maximum value for which the same output time by the earliest time is accomplished. The latest starting times of any of the processes are given by [8]:

$$\boldsymbol{x}_L(k)=(\boldsymbol{A}_k^0\boldsymbol{F}_k^*)^T \odot \boldsymbol{x}(k+1) \wedge (\boldsymbol{C}_k^0\boldsymbol{F}_k^*)^T \odot \boldsymbol{y}(k). \quad (20)$$

The latest feeding times are also given by:

$$\boldsymbol{u}_L(k) = (\boldsymbol{B}^0)^T \odot \boldsymbol{x}_L(k). \quad (21)$$

A critical path is defined as the processes whose total floats are zero. Moreover, the total float is defined as a total sum of the float times of the corresponding processes. It is also stated as the difference between two primary times, one of which is the minimum value among the latest starting times of the succeeding processes, by which the output time is unchanged, and the other of which is the completion time in the corresponding process caused by the earliest starting time. The total floats regarding all processes are obtained as:

$$\boldsymbol{w}(k) = \boldsymbol{x}_L(k) - \boldsymbol{x}_E(k). \quad (22)$$

The critical path is determined by a set of numbers α that satisfy:

$$\{\ \alpha \mid [\boldsymbol{w}(k)]_\alpha = 0\}. \quad (23)$$

3 Critical Chain Project Management

We briefly review the concept of the critical chain project management (CCPM) method. Projects often exceed their initial planned schedule. This is usually due to unforeseen uncertainties related to external factors. To resolve this dilemma, the CCPM method is often considered (e.g. [5] and the references therein). 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 linked processes in the entire project. Focusing only on the longest chain of processes may result in several problems, such as multitasking. The CCPM method instead asserts that, in addition to process dependencies, a good project management should address resource constraints only if they are absolutely required. The CCPM method provides a method for determining places locations at which time buffers should be inserted in order to prevent unplanned delays in completing the project. In the PERT, each process in the project consists of a set of four times: the earliest start time, the earliest output time, the latest start time, and the latest output time. Since these times are made known to everyone involved in the project, they can be closely monitored. The difference between the earliest and the latest start times is equivalent to the slack. Processes on the critical path do not have any slack time and should be given significant attention. In estimating the process duration, there is a tendency to use optimistic estimates, which include significant safety margins, in order to ensure the completion in the specified time frame. This is often referred to as the 90% estimate. In the CCPM method, an empirical value is used to obtain an estimate of the Aggressive But Possible (ABP) time for each process, and this value is used for the process duration. In the present paper, we use $\mathrm{ABP} = \mathrm{HP}^{\otimes\frac{1}{3}}$, where Highly Possible (HP) is the time to completion with a probability of 90%.

The next step is to determine a buffer to encapsulate the uncertainty of task durations. This is referred to as the project buffer, and it absorbs variations in the critical path. A project buffer is embedded between the final process on the critical path and the external output. A feeding buffer is inserted before a process that joins the critical path but is not on the critical path. The role of the feeding buffer is to protect the critical chain from variations in non-critical paths of tasks, i.e., to help maintain the critical path as the critical path.

Finally, the critical path is monitored by closely monitoring the rate at which the project buffer is consumed.

4 Application of a CCPM Based Framework on MPL System

In this section, we briefly review how to determine the position and the size of the Project Buffer (PB) and Feeding Buffer (FB). Many more details of the method for applying the concepts of CCPM on MPL systems were discussed in [1]. Before the discussion about determination of the position and size of buffer, we introduce the following two vectors :

$$[\boldsymbol{v}]_i \equiv \begin{cases} e & : i \in \beta \\ \varepsilon & : i \notin \beta \end{cases}, [\boldsymbol{w}]_i \equiv \begin{cases} e & : i \in \alpha \\ \varepsilon & : i \notin \alpha \end{cases}, \quad (24)$$

where β is a set of processes on the non-critical path. Moreover, the adjacency matrix $\boldsymbol{F}_k$ is decomposed using the following formula:

$$\boldsymbol{F}_k = \boldsymbol{F}_{\alpha\alpha} \oplus \boldsymbol{F}_{\alpha\beta} \oplus \boldsymbol{F}_{\beta\alpha} \oplus \boldsymbol{F}_{\beta\beta}, \quad (25)$$

where

$$\begin{aligned} \boldsymbol{F}_{\alpha\alpha} &\equiv \mathrm{diag}(w)^T \otimes \boldsymbol{F}_k \otimes \mathrm{diag}(w), & (26)\\ \boldsymbol{F}_{\alpha\beta} &\equiv \mathrm{diag}(v)^T \otimes \boldsymbol{F}_k \otimes \mathrm{diag}(w), & (27)\\ \boldsymbol{F}_{\beta\alpha} &\equiv \mathrm{diag}(w)^T \otimes \boldsymbol{F}_k \otimes \mathrm{diag}(v), & (28)\\ \boldsymbol{F}_{\beta\beta} &\equiv \mathrm{diag}(v)^T \otimes \boldsymbol{F}_k \otimes \mathrm{diag}(v). & (29) \end{aligned}$$

$\boldsymbol{F}_{\alpha\alpha}$ means the adjacency matrix that shows the transition from the process on the critical path onto the process on the critical path. Similarly, $\boldsymbol{F}_{\alpha\beta}$ means the adjacency matrix that shows the transition from the process on the critical path onto the process on the non-critical path. $\boldsymbol{F}_{\beta\alpha}$ is the adjacency matrix that shows the transition from the process on the non-critical path onto the process on the critical path. $\boldsymbol{F}_{\beta\beta}$ is also the adjacency matrix that shows the transition from the process on the non-critical path onto the process on the non-critical path.

4.1 Position and Size of Project Buffer

In order to prevent delay of the project, the PB is embedded. The PB is embedded between the final process on the critical path and the external output, and its position is determined by using the output matrix $[\boldsymbol{C}_k^0]_{ij}$. If the element $[\boldsymbol{C}_k^0]_{i\alpha}$ has a finite value, process i has an external output, then we embed the PB after process i. The size of PB is estimated following formula,

$$\mathrm{PB} = \sigma_E^{\otimes\frac{1}{3}}, \quad (30)$$

where $\sigma_E = (\boldsymbol{y}_E^T \otimes \boldsymbol{y}_E)^{\otimes\frac{1}{2}}$. This method is based on "cut and paste method" or "50% of the chain" [5].

ISSN: 1790-2769

ISBN: 978-960-474-168-7

4.2 Position and Size of Feeding Buffer

In order to protect a critical path from delay of a non-critical path, the FB is embedded. From the discussion in Sec.3, the FB is embedded in the juncture between non-critical path and critical path. The position of FB is determined by :

$$[\boldsymbol{v'}]_\gamma = \left[\boldsymbol{F}_{\beta\alpha}^T \otimes \boldsymbol{w}\right]_\gamma . \quad (31)$$

If the γ-th element of Eq.(31) has a finite value, a FB is embedded after process γ.

Next, we discuss how to determine the size of the FB. We basically consider the size of FB to sum up the processing times of processes on the non-critical path: from the fork between the critical path and the non-critical path to the joint between the critical path and the non-critical path. The size of FB is estimated by [1]:

$$\text{FB} \equiv \left[\text{diag}(\boldsymbol{v'}) \otimes \tilde{F}_{\beta\beta}^* \otimes \boldsymbol{v}\right]^{\otimes\frac{1}{3}} , \quad (32)$$

where $\tilde{\boldsymbol{F}}_{\beta\beta} \equiv \text{diag}(\boldsymbol{j}^\sharp) \otimes \boldsymbol{F}_{\beta\beta}$ means that only the transition from the process where FB is embedded to the process in β has been deleted. $\boldsymbol{j} \equiv [\boldsymbol{F}_{\beta\beta} \otimes \boldsymbol{v'}]^{\otimes\frac{1}{2}}$. If the γ-th element of $\boldsymbol{j}$ has finite value, then the process γ is on the non-critical path and the process γ also has FB in preceding process. An operator $\sharp$ for a vector $\boldsymbol{p}$ defined as :

$$\{[\boldsymbol{p}]_i\}^\sharp \equiv \begin{cases} \varepsilon : \text{if } i\text{-th element is a finite value.} \\ e : \text{if } i\text{-th element is } \varepsilon. \end{cases} \quad (33)$$

5 Toward the Resolution of Resource Conflict

In this section, toward the resolution of resource conflict, we propose a method for discovering the resource conflict between two different projects in the MPL-CCPM representation. We also discuss how to calculate the amount of shift in order to solve resource conflict.

5.1 Proposed Method

5.1.1 Discovery of the Resource Conflict

We discuss how to discover the resource conflict between project A and B.

For the project A in process i, let $[\boldsymbol{x}_A^-]_i$ and $[\boldsymbol{x}_A^+]_i$ be the earliest starting time and the earliest completion times, respectively. Similarly, for the project B in process i, let $[\boldsymbol{x}_B^-]_i$ and $[\boldsymbol{x}_B^+]_i$ be the earliest starting time and the earliest completion times, respectively. We assume that we use the same resource, if the process number between the project A and B are same. In order to discover the resource conflict between the project A and B, we calculate :

$$\boldsymbol{c} = \left(\text{diag}\left(\boldsymbol{x}_B^-\right) \odot \boldsymbol{x}_A^+\right) \wedge \left(\text{diag}\left(\boldsymbol{x}_A^-\right) \odot \boldsymbol{x}_B^+\right) . \quad (34)$$

If $[\boldsymbol{c}]_i > 0$, the resource conflict is occurred in process i.

5.1.2 Shift of the Resource

In this subsection, we discuss how to calculate the amount of the shift in order to solve resource conflict between two projects.

In order to solve the resource conflict, we need to shift the resource in the projects. Before the discussion about the shift, we should determine which projects have high priority. If the resource between two projects is conflict, the policy to solve the resource conflict is as follows : by the definition of CCPM method [5], we should not delay the project which has high priority. Thus, when the resource between two projects is conflict, we should assign the resource to the project which has high priority.

Now, we assume that the resource s between project A and B is conflict : i.e. $[c]_s > 0$. We also assume that the project A has higher priority than that of project B. From the above discussion, we should not delay the project A. Therefore, after the shift, the process s in the project A must start earlier than that of the project B. In this paper, for simplicity, after the shift, we assume that when the process k of the project A is completed, the process k of the project B starts immediately. In this case, in order to solve resource conflict, we should shift the process k in the project B backward. We find that the amount of shift is $[\boldsymbol{x}_A^+]_s - [\boldsymbol{x}_B^-]_s$. From the above discussion, we can get a general formulation to calculate the amount of shift $[\boldsymbol{d'_m}]_i$ in order to solve resource conflict is given by :

$$[\boldsymbol{d'_m}]_i = \left[\text{diag}\left(\boldsymbol{x}_B^-\right) \odot \boldsymbol{x}_A^+\right]_i . \quad (35)$$

6 Numerical Example

In this section, a simple model and numerial example are presented to promote a better understanding of the proposed method in section 5.

The left side of Fig.2 shows simple two projects, called A and B, which have with one-input, one-output, two processes. The right side of Fig.2 shows the gantt chart of these projects.

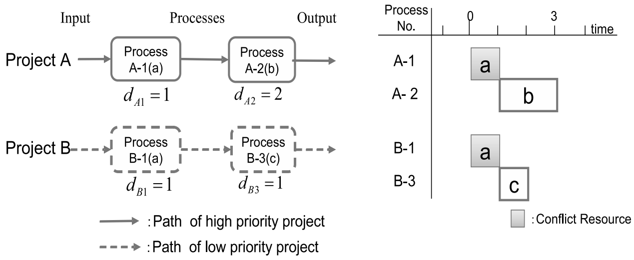

Figure 2: The two projects with resource conflict.

We assume that the project A has higher priority than that of project B. We can calculate the starting times $\boldsymbol{x}_A^-$ and the completion times $\boldsymbol{x}_A^+$ in project A :

$$\boldsymbol{x}_A^-=\begin{pmatrix} e & \varepsilon & 1\end{pmatrix}^T, \boldsymbol{x}_A^+=\begin{pmatrix} 1 & 3 & \varepsilon\end{pmatrix}^T. \tag{36}$$

Similarly, we can also calculate $\boldsymbol{x}_B^-$ and $\boldsymbol{x}_B^+$ in project B :

$$\boldsymbol{x}_B^-=\begin{pmatrix} e & \varepsilon & 1\end{pmatrix}^T, \boldsymbol{x}_B^+=\begin{pmatrix} 1 & \varepsilon & 2\end{pmatrix}^T. \tag{37}$$

Using Eq.(34), the resource conflict between the projects A and B can be discovered. Eq.(34) are calculate using Eqs. (36) and (37) :

$$\boldsymbol{c}=\begin{pmatrix} 1 & \varepsilon & \varepsilon\end{pmatrix}. \tag{38}$$

Because of $[\boldsymbol{c}]_1 > 0$, we can find that the resource "a" of the process A-1 in project A is in conflict with the resource "a" of the process B-1 in project B.

Using Eq.(35), we can calculate the amount of the shift :

$$\boldsymbol{d'_m}=\begin{pmatrix} 1 & \varepsilon & \varepsilon\end{pmatrix}^T. \tag{39}$$

Then, we can find that we should shift the process B-1 backward and the amount of shift is 1, in order to solve the resource conflict.

7 Conclusion

In this paper, we have reviewed a method for applying the concept of the feeding and project buffers in the CCPM-MPL method in order to address the project even if an undesirable state change has occurred. We have also proposed a method for discovering the resource conflict between two different projects in the MPL-CCPM representation. As a result, we realized a method for discovering the resource conflict between two different projects.

The proposed method is useful to discover the resource conflict between two different projects. However, we need to discuss the method for rescheduling in order to resolution of resource conflict. We will also discuss a method for discovering the resource conflict in the same project. These extensions will discussed elsewhere.

Acknowledgements: Hirotaka Takahashi was supported in part by the Uchida Energy Science Promotion Foundation, Sasagawa Scientific Research Grand and JSPS Grant-in-Aid for Scientific Research No.20540260. Hiroyuki Goto was supported in part by research grants from Takahashi Industrial and Economic Research Foundation, the Mazda Foundation, the Kayamori Foundation of Informational Science, and the JSPS Grants-in-Aid for Scientific Research Nos. 20710114 and 20560423.

References:

[1] H. Takahashi, H. Goto and M. Kasahara, Application of A Critical Chain Project Management Based Freamework on Max-plus Linear Systems, *International Journal of Computational Science*, Vol.3, 2009, pp.117–132.; H. Takahashi, H. Goto, M. Kasahara, Toward the Application of a Critical-Chain-Project-Management-based Freamework on Max-plus Linear Systems, *Industrial Engineering and Management Systems*, vol.8, 2009, pp155–161.

[2] G. Cohen, P. Moller, J. Quadrat, and M. Viot, Algebraic tools for the performance evaluation of discrete event systems, *Proceedings of the IEEE*, Vol.77, 1989, pp. 39–59.; B. Heidergott, G. J. Olsder, and L. Woude, *Max Plus at work: Modeling and Analysis of Synchronized Systems.* New Jersey, Princeton University Press. , 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*, New York : Springer Verlag, 2006.

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

[6] E. M. Goldratt, *Theory of Constraints : And how it should be implemented*, North River Pr., 1990.

[7] M. Kasahara, H. Takahashi and H. Goto, On a Buffer Management Policy for CCPM-MPL Representation, *International Journal of Computational Science*, Vol.4, 2010 (To appear).

[8] H. Goto and S. Masuda, Monitoring and scheduling methods for MIMO-FIFO systems utilizing max-plus linear representation, *Industrial Engineering and Management Systems*, Vol.7, 2008, pp. 23–33.