Industrial Engineering
& Management Systems
Vol 11, No 4, December 2012, pp.397-405
ISSN 1598-7248 │EISSN 2234-6473 │

http://dx.doi.org/10.7232/iems.2012.11.4.397

Buffer Management Method for Multiple Projects in the CCPM-MPL Representation

Nguyen Thi Ngoc Truc*
Department of Information Science and Control Engineering, Nagaoka University of Technology, Nagaoka, Japan

Yoshinori Takei
Department of Electrical Engineering, Nagaoka University of Technology, Nagaoka, Japan

Hiroyuki Goto
Department of Industrial and Systems Engineering, Hosei University, Koganei, Japan

Hirotaka Takahashi
Department of Humanities, Yamanashi Eiwa College, Kofu, Japan

(Received: August 19, 2012 / Revised: November 23, 2012 / Accepted: November 26, 2012)

ABSTRACT

This research proposes a framework of buffer management for multi-project systems in the critical chain project management (CCPM) method, expressed in the form of max-plus linear (MPL) representation. Since time buffers are inserted in the projects for absorbing uncertainties in task durations and protecting the completion times, the proposed method provides a procedure for frequently surveying the rates of consumed buffers and the rate of elapsed times. Their relation expresses the performance of the projects which is plotted on a chart through the completed processes. The chart presents the current performance of the projects and their interaction, which alerts managers to make necessary decisions at the right time for managing each project and the entire multi-project system. The proposed framework can analyze the complex system readily, and it enables managers to make an effective decision on scheduling. The effectiveness of the framework is demonstrated through a numerical example.

Keywords: Max-Plus Linear System, Critical Chain Project Management, Time Buffers, Multiple Projects, Buffer Management, Fever Chart

* Corresponding Author, E-mail: s107009@stn.nagaokaut.ac.jp

1. INTRODUCTION

The critical chain project management (CCPM) is a well-known method for planning and managing projects. The concept of the CCPM is an outgrowth of the theory of constrains (TOC), first developed by Goldratt (1990), which has drawn much attention for many researchers (Herroelen *et al.*, 2002; Cohen *et al.*, 2004; Leach, 2005; Tukel *et al.*, 2006). These researches have mainly focused on clarifying the concepts of the CCPM method, discussing several related issues such as resource conflict, buffer's size, and buffer management. In project management, initial schedule is frequently changed due to unpredictable reasons such as external uncertainties, processing changes, and resource conflicts. Thus, the CCPM provides a method for inserting time buffers which play a role for absorbing uncertainties in the task durations to protect the completion time. Moreover, the CCPM method has achieved successes not only for a single project but for large-scale multi-projects (Cohen *et al.*, 2004; Leach, 2005).

On the other hand, the max-plus linear (MPL) representation is known as an efficient tool for describing a class of discrete event systems (DESs) (Heidergott *et al.*, 2006; Goto, 2010). The typical and significant feature of DESs is that events occur at discrete time instants and the values of the internal states change non-continuously. This kind of systems typically appears in manufacturing systems, transportation systems, project management, and so on. Since unpredictable changes in the execution times may influence the completion time, several researches that consider uncertainties have been carried out (Heidergott, 2006). However, due to the nonlinearity in the states of the system, it is difficult to handle large-scale systems.

Furthermore, an application of the CCPM framework to an MPL representation was studied in our previous paper (Takahashi *et al.*, 2009; Truc *et al.*, 2011). Therein, a method for determining and inserting time buffers to absorb uncertainties is considered using an MPL representation. We first identified a critical path, which is the longest path of successive tasks with zero float time. Then, we reduced the processing time and installed project and feeding buffers. A project buffer is inserted between the end of the critical path and its succeeding output. Feeding buffers are inserted wherever a non-critical path joins into a critical path. These buffers were frequently monitored and controlled with a buffer management policy (Kasahara *et al.*, 2009), in which the behavior of a project is featured by the rates of buffer consumption and processing time. However, these works were designed only for a single-project system.

Recently, a CCPM-MPL framework for determining time buffers has been proposed for multi-project systems (Truc *et al.*, 2012), an extended and improved version of a single-project case. The project and feeding buffers are determined for each project in large-scale systems; the buffers are used to absorb uncertainties in the internal task durations of each project. Moreover, the constraints between the projects are determined to insert capacity buffers; they play a role for absorbing uncertainties in the preceding project and protect the subsequent projects.

This research considers a framework for managing these buffers in a multi-project system. Particularly, instead of minimizing the duration of each project as the case of a single project, the multi-project scheduling aims to maximize the throughput of the entire system by sharing the same resources. Thus, it is difficult for managers to comprehend the relationship between these projects with mutual dependence by applying the single-project framework (Kasahara *et al.*, 2009) one by one, making the control of the whole system tough. The proposed framework herein expresses the performance for all projects and their relationship in the complex system with multiple input times, which can alert managers to make extra actions at the proper time to avoid tardiness both for individual project and for the entire system. In this paper, the effect of cost and resource conflicts is not considered for the sake of simplicity. For further discussion of resource conflicts, refer to Yoshida *et al.* (2011).

The remaining contents are presented in the following four sections. Section 2 provides a brief review of the theoretical background for this paper: max-plus algebra, MPL representation, and CCPM-MPL framework. Section 3 derives a concept of buffer management in the CCPM method, and proposes a framework of buffer management in the form of the MPL representation. The proposed framework is confirmed through a numerical example in Section 4. Finally, conclusions and recommendations are given in Section 5.

2. THEORETICAL BACKGROUND

2.1 Max-Plus Algebra

Max-plus algebra is known as the schedule algebra for describing a certain class of DESs. For a set $D = R$ $\{-\infty\}$, where R is the entire real set, operators for addition and multiplication are defined as:

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

The symbol $\otimes$ corresponds to multiplication in conventional algebra, and is often omitted when no confusion is likely to arise. For example, we simply write xy for expressing $x \otimes y$. These operators hold the commutative, associative and distributive laws. The zero and unit elements for these are given by $\varepsilon(=-\infty)$ and e $(=0)$, respectively. These hold $x \oplus \varepsilon = \varepsilon \oplus x = \varepsilon$ and x $\otimes \varepsilon = \varepsilon \otimes x = \varepsilon$ for an arbitrary $\bar{\varepsilon}(=+\infty)$. Moreover, we define $\bar{\varepsilon}(=+\infty)$ and assume a property $\varepsilon \otimes \varepsilon = \varepsilon \otimes \varepsilon = \varepsilon$.

Furthermore, the following two operators are defined for subsequent discussions:

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

An operator for the power of $a \in R$ is defined as:

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

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

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

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

For a matrix $X \in D^{m \times n}$, $[X]i_j$ expresses the $(i, j)^{\text{th}}$ element of X, and X^T is the transpose matrix of X. For $X, Y \in D^{m \times n}$,

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

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

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

$$[\boldsymbol{X} \oplus \boldsymbol{Y}]_{ij} = \oplus_{k=1}^{l}\left([\boldsymbol{X}]i_j,\ [\boldsymbol{Y}]i_j\right) \tag{8}$$

$$[\boldsymbol{X} \odot \boldsymbol{Y}]_{ij} = \wedge_{k=1}^{l}\left([\boldsymbol{X}]i_j,\ [\boldsymbol{Y}]i_j\right) \tag{9}$$

The priority of operators $\otimes$ and $\odot$ are higher than that of operators $\oplus$ and $\wedge$ where $\odot$ is well-defined if $\boldsymbol{X}$ has at least one non-ε entry in every row. The zero and unit elements for matrices are; ε_{mn} is a matrix whose all elements are ε in $D^{m\times n}$, and $\boldsymbol{e}$ is a matrix whose diagonal elements are e and off-diagonal elements are ε. For a vector $\boldsymbol{v} \in D^m$, $\mathrm{diag}(\boldsymbol{v}) \in D^{m\times n}$ represents a matrix whose diagonal elements are $[\boldsymbol{v}]_i$ and off-diagonal elements are ε.

2.2 Max-Plus Linear Representation

We briefly review the MPL representation for a certain class of discrete event systems developed by Yoshida *et al.* (2010). We assume the following constraints 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.
- The subsequent job cannot start processing while the current job is at work.
- Processes that have precedence constraints cannot start processing until they have received all the required parts from the preceding processes.
- Processes with external inputs cannot start until all the required materials have arrived.
- The processing starts as soon as all the conditions above are satisfied.

For the k^{th} job in process $i\,(1 \le i \le n)$, let $d_i(k)(\ge 0)$, $\left[\boldsymbol{x}^-(k)\right]_i$, and $\left[\boldsymbol{x}^+(k)\right]_i$, be the processing time, processing start time, and processing completion time, respectively. We give the initial condition by $\boldsymbol{x}(0) = \varepsilon_{n1}$. Moreover, $[\boldsymbol{u}(k)]_i$ represents the material arrival time from external input $i\,(1 \le i \le p)$, and $[\boldsymbol{y}(k)]_i$ represents the output time of the finished product to external output $i\,(1 \le i \le q)$. The following matrices $\boldsymbol{P}_k$, $\boldsymbol{F}_0$, $\boldsymbol{B}_0$ and $\boldsymbol{C}_0$ are introduced for representing the structure of the system:

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

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

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

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

where $\boldsymbol{F}_0$, $\boldsymbol{B}_0$ and $\boldsymbol{C}_0$ are referred to as the adjacency, input and output matrices, respectively.

The earliest completion time is defined as the minimum value with which the corresponding process can complete processing. Specifically, the earliest completion times of all processes are calculated by:

$$x_E^+(k) = (\boldsymbol{P}_k\,\boldsymbol{F}_0)^* + \boldsymbol{P}_k[\boldsymbol{x}^+(k-1) \oplus \boldsymbol{B}_0\boldsymbol{u}(k)], \tag{10}$$

where $(\boldsymbol{P}_k\,\boldsymbol{F}_0)^*\boldsymbol{e}_n \oplus \boldsymbol{P}_k\,\boldsymbol{F}_0 \oplus \cdots \oplus (\boldsymbol{P}_k\boldsymbol{F}_0)^{l-1}$, and an instance $l\,(1 \le l \le n)$ depends on the precedence-relationships of the system. The corresponding output times are given by:

$$\boldsymbol{y}_E(k) = \boldsymbol{C}_0 \otimes \boldsymbol{x}_E^+(k) \tag{11}$$

The earliest starting times of all processes are given by:

$$\boldsymbol{x}_E^-(k) = \boldsymbol{P}_k \odot \boldsymbol{x}_E^+(k). \tag{12}$$

Furthermore, the latest starting times are defined as the maximum value by which the same output time based on the earliest time can be accomplished. Thus, the latest starting times of all processes and the latest feeding times are given by:

$$\boldsymbol{x}_L^-(k) = \left[(\boldsymbol{P}_k\boldsymbol{F}_0)^*\boldsymbol{P}_k\right]T \odot \left[C_0^T \odot \boldsymbol{y}_E(k)\right] \tag{13}$$

$$\boldsymbol{u}_L(k) = \boldsymbol{B}_0^T \odot \boldsymbol{x}_L^-(k) \tag{14}$$

A critical path is understood as a series of processes with zero total float. Total float is the maximum time which can be moved backward without changing the completion time. This is defined as the difference between the latest and earliest starting times. Specifically, the total floats $\boldsymbol{\omega}(k)$ of all processes are obtained as:

$$\boldsymbol{\omega}(k) = \mathrm{diag}\left(\boldsymbol{x}_E^-(k)\right) \odot \boldsymbol{x}_L^-(k) \tag{15}$$

Moreover, the critical path is identified by the set of process numbers α that satisfy:

$$\left\{\alpha \middle| [\boldsymbol{\omega}(k)]_\alpha = 0\right\} \tag{16}$$

2.3 CCPM-MPL Framework for Determining Time Buffers in Multiple Projects

We briefly review the framework for determining time buffers in a multi-project system (Truc *et al.*, 2012). Considering a general system with m processes (tasks) and n projects, n vectors $\boldsymbol{l}_1, \boldsymbol{l}_2, \cdots, \boldsymbol{l}_n$ of m-dimension are introduced with the following properties:

$$\left[\boldsymbol{l}_j\right]_i = \begin{cases} e : \text{if process } i \text{ belongs to project } j, \\ \varepsilon : \text{otherwise} \end{cases} \tag{17}$$

for all $i\,(1 \le i \le m)$ and $j\,(1 \le j \le n)$, and $\boldsymbol{l}_j$ expresses which processes belong to project j. Note that process i is implicitly designed to belong a certain project based on the priority of the projects and the processes. Then, we define a matrix $\boldsymbol{L} \in D^{m \times n}$ that represents the layer of projects as follows:

$$\boldsymbol{L} = [\boldsymbol{l}_1 \, \boldsymbol{l}_2 \cdots \boldsymbol{l}_n] \tag{18}$$

Moreover, we reduce the processing time of tasks to 1/3 of the original time and define the size of time buffers as 1/3 of the corresponding path duration (Takahashi *et al.*, 2009). The position and size of the buffers are determined as below:

- **Project buffer (PB)**

Project buffers are inserted between the end of critical paths and their succeeding external outputs, to protect the critical paths. The positions are determined by inspecting the elements of the output matrix $\boldsymbol{C}_0$, if $[\boldsymbol{C}_0]_{ij} \neq \varepsilon$ and $j \in \alpha$, a PB is inserted behind process. The sizes of the PBs are obtained by:

$$\boldsymbol{b}_p = (\mathrm{diag}(\boldsymbol{L}_\alpha^T \odot \boldsymbol{x}_L^-) \odot \boldsymbol{y}_E)^{\otimes \frac{1}{3}} \tag{19}$$

where matrix $\boldsymbol{L}_\alpha$ is defined from the set of critical processes (α) determined in Eq. (16) as:

$$[\boldsymbol{L}_\alpha]_{ij} = \begin{cases} e : \text{if } i \in \alpha \text{ and} [L]_{ij} = e, \\ \varepsilon : \text{otherwise} \end{cases} \tag{20}$$

- **Feeding buffer (FB)**

Feeding buffers are inserted where a non-critical path joins into a critical path, to protect the critical paths from delay which may occur in the non-critical paths. To define the positions of FBs, we first introduce two vectors $\boldsymbol{v}$ and $\boldsymbol{w}$ with the properties as $[\boldsymbol{v}]_i = \{e : i \in \beta, \varepsilon : i \notin \beta\}$, and $[\boldsymbol{w}]_i = \{e : i \in \alpha,\ \varepsilon : i \notin \alpha\}$, where β is the set of non-critical processes, the positions are identified using two vectors $\boldsymbol{v}'$ and $\boldsymbol{w}'$ with the following properties:

$$[\boldsymbol{v}']_i = [\boldsymbol{F}_{\beta\alpha}^T \otimes \boldsymbol{w}]_i = \begin{cases} e : \text{if } i \in \gamma, \\ \varepsilon : \text{otherwise,} \end{cases} \tag{21}$$

where γ is the subset of β having a successor classified as α. $\boldsymbol{F}_{\beta\alpha} = \mathrm{diag}(\boldsymbol{w}) \otimes \boldsymbol{F}_0 \otimes \mathrm{diag}(\boldsymbol{v})$, $\boldsymbol{F}_{\beta\alpha}$ represents the transitions from non-critical processes to critical ones.

$$[\boldsymbol{w}']_i = [\boldsymbol{F}_{\beta\alpha} \otimes \boldsymbol{v}]_i = \begin{cases} e : \text{if } i \in \tau, \\ \varepsilon : \text{otherwise,} \end{cases} \tag{22}$$

where τ is the subset of α having a predecessor classified as β. Then, FBs are installed at transitions between processes γ and τ. The sizes of the FBs are estimated as:

$$\boldsymbol{b}_f = (\mathrm{diag}(\boldsymbol{P}_k \otimes \boldsymbol{v}') \otimes (\boldsymbol{F}_{\beta\beta} \otimes \boldsymbol{P}_k)^* \otimes \boldsymbol{v})^{\otimes \frac{1}{3}} \tag{23}$$

where $\boldsymbol{F}_{\beta\beta} = \mathrm{diag}(v) \otimes \boldsymbol{F}_0 \otimes \mathrm{diag}(\boldsymbol{v})$, $\boldsymbol{F}_{\beta\beta}$ represents the transitions from non-critical processes to non-critical ones.

- **Capacity buffer (CB)**

Capacity buffers are inserted at the transitions between projects, to absorb uncertainties in the preceding project and protect the subsequent projects. We first determine transitions for each pair of the projects. Generally, transitions from project A to project B(A ≠ B) are defined through two vectors $\boldsymbol{h}$ and $\boldsymbol{h}'$ with the following properties as:

$$[\boldsymbol{w}']_i = [\boldsymbol{F}_{\beta\alpha} \otimes v]_i = \begin{cases} e : \text{if } i \in \tau, \\ \varepsilon : \text{otherwise,} \end{cases} \tag{24}$$

where ϑ is the set of processes in project A with a successor belonging to project B. $\boldsymbol{F}_{AB} = \mathrm{diag}(\boldsymbol{l}_B) \otimes \boldsymbol{F}_0 \mathrm{diag}$ $(\boldsymbol{l}_A)$, $\boldsymbol{F}_{AB}$ represents the transitions from processes in project A to processes in project B.

$$[\boldsymbol{hw}]_i = [\boldsymbol{F}_{AB} \otimes \boldsymbol{l}_A]_i = \begin{cases} e : \text{if } i \in \varphi, \\ \varepsilon : \text{otherwise,} \end{cases} \tag{25}$$

where φ is the set of processes in project B with a predecessor belonging to project A. Taking these together, CBs are inserted at transitions between processes ϑ and φ. The sizes of the CBs from project A to B are calculated by:

$$\boldsymbol{b}_c = (\mathrm{diag}(\boldsymbol{P}_k \otimes \boldsymbol{h}) \otimes (\boldsymbol{F}_{AA} \otimes \boldsymbol{P}_k) * \otimes \boldsymbol{l}_A)^{\otimes \frac{1}{3}} \tag{26}$$

where $\boldsymbol{F}_{AA} = \mathrm{diag}(\boldsymbol{l}_A) \otimes \boldsymbol{F}_0 \otimes \mathrm{diag}(\boldsymbol{l}_A)$. $\boldsymbol{F}_{AA}$ represents the transitions between processes in project A.

Similarly, we can find the positions and sizes of CBs from project B to A, or between any pair of the projects.

- **Redefined system with the CCPM applied**

The representing matrices introduced in Section 2.2 are redefined for easily comprehending the structure of the system after the CCPM applied. After the insertion of the PB, the output matrix $\boldsymbol{C}_\gamma$ is reformulated in the following manner:

$$\boldsymbol{C}_\gamma = \mathrm{diag}(\boldsymbol{b}_p) \otimes \boldsymbol{C}_0, \tag{27}$$

where the subscript ‘ γ ’ expresses that the matrix was redefined. After the insertion of the FBs and CBs, the adjacency matrix $\boldsymbol{F}_\gamma$ is reformulated as:

$$\boldsymbol{F}_\gamma = \boldsymbol{F}_0 \oplus \bar{\boldsymbol{F}}_{\beta\alpha} \oplus \boldsymbol{F}_c, \tag{28}$$

where $\boldsymbol{F}_{\beta\alpha} = \boldsymbol{F}_{\beta\alpha} \otimes \mathrm{diag}(\boldsymbol{b}_f) : \boldsymbol{F}_{\beta\alpha}$ and $\boldsymbol{b}_f$ expresses the position and sizes of the FBs, respectively, and

$$\boldsymbol{F}_c = \bigoplus_{\substack{x,y=1 \\ x \neq y}}^{n} \Big(\boldsymbol{F}_{xy} \otimes \mathrm{diag}\big((\boldsymbol{b}_c)_{xy}\big)\Big), \tag{29}$$

where n is the number of projects. The subscript 'c' expresses the CBs. $\boldsymbol{F}_{xy}$ presents the transitions of processes from project x to y, and $(\boldsymbol{b}_c)_{xy}$ expresses the size of the corresponding CBs. Note that $\boldsymbol{F}_c = \varepsilon_{mn}$ if n = 1. Moreover, matrices $\boldsymbol{P}_\gamma = \boldsymbol{P}_k^{\otimes 1/3}$ and $\boldsymbol{B}_0$ remain unchanged.

3. BUFFER MANAGEMENT

3.1 Concept of the CCPM

In the CCPM, for managing and observing the projects' performance, the buffers are frequently monitored to protect the critical path as well as the project's completion time. This mechanism is called "Buffer Management", which can detect a potential problem and raise a warning signal to managers. As the projects proceed, if a task elapses a longer time than expected, the task consumes the buffer on the corresponding path. Thus, the buffer management frequently compares two parameters: the consumption rate of the PB and the progress rate of the corresponding critical path, for expressing the current performance of the project. In the multi-project system, these parameters are frequently checked for all projects and plotted on a fever chart, as shown in Figure 1. This chart shows the relationship between buffer used (%) versus time used (%) through the completed processes. The chart is divided into three zones: green, yellow, and red, which are determined empirically by thresholds settings. Typically, we base on the thresholds settings proposed by Leach (2005) as listed in Table 1.

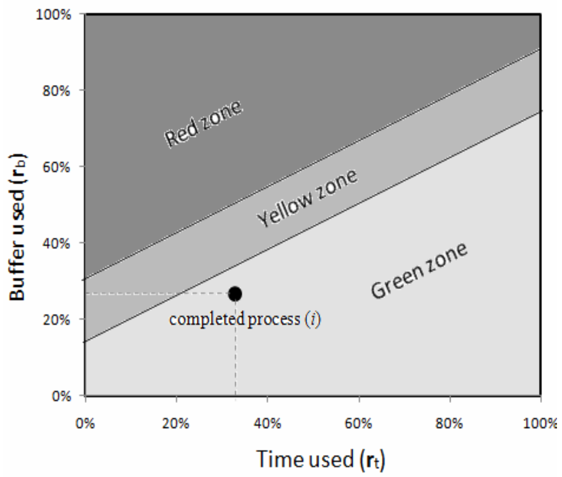

Figure 1. An example of a fever chart.

Table 1. Buffer thresholds

Time used (%)	Buffer used (%)	
	Green to yellow transition (%)	Yellow to red transition (%)
0	15	30
100	75	90

Using the fever chart, the projects' status is alerted to project managers and the decision level for an extra action is suggested. Specifically, if the current status is in the green zone, the project is going well and the managers need not take an action. If the status is in the yellow zone, the project assesses a problem and needs a recovery plan to avoid further buffer erosion. If the status is in the red zone, the project will possibly be late and the managers should initiate the action.

3.2 Proposed Method

We develop a framework of buffer management for a multi-project system, which focuses on the system after the insertion of the time buffers. The overall procedure for buffer management is summarized as the flowchart in Figure 2. A sequencing description for the procedure is expressed by using the MPL representation as follows:

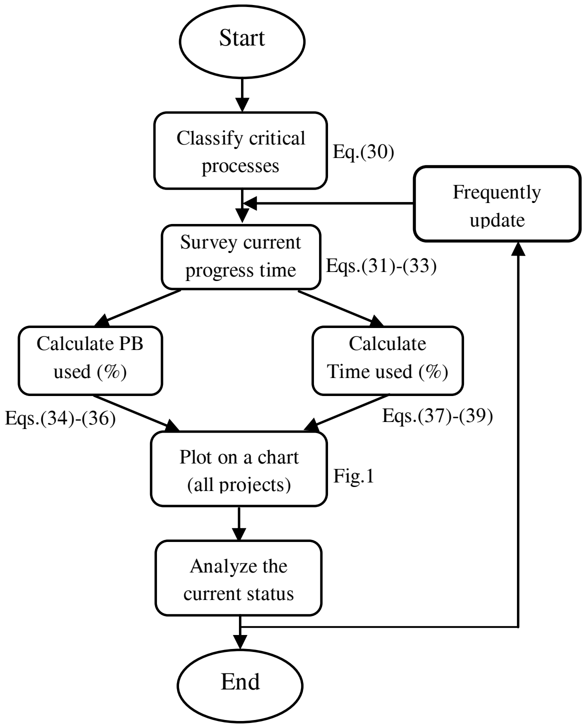

Figure 2. Flowchart of buffer management.

We again consider the multi-project system with n projects and m processes mentioned in Section 2.3. Using n vectors ($\boldsymbol{l}_1, \boldsymbol{l}_2, \cdots, \boldsymbol{l}_n$) which represent the processes belonging to n projects, derived in Eq. (17), we define n vectors $\boldsymbol{l}_{1\alpha}, \boldsymbol{l}_{2\alpha}, \cdots, \boldsymbol{l}_{n\alpha}$ that satisfy:

$$[\boldsymbol{l}_{x\alpha}]_i = \begin{cases} e : if\ i \in \alpha \text{ and } [\boldsymbol{l}_{x\alpha}]_i = e, \\ \varepsilon : \text{otherwise}, \end{cases} \tag{30}$$

where α is the set of critical processes determined by Eq. (16), and $\boldsymbol{l}_{x\alpha}$ expresses which critical processes belong to project $x (1 \le x \le n)$. Thus, the managers can monitor each project in only one procedure for the entire system instead of considering them independently. Next, we introduce a vector $\boldsymbol{z}$ which represents the actual completion times of the completed processes at the monitoring stage. Thus, the managers can survey the value of this vector frequently whenever monitoring the project's execution. Vector $\boldsymbol{z}$ has the following property:

$$[z]_i = [\boldsymbol{F}_{\beta\alpha} \otimes v]_i = \begin{cases} t_i : \ i \in \eta, \\ \bar{\varepsilon} : \text{otherwise}, \end{cases} \tag{31}$$

where t_i is defined as actual completion time of process i and η represents the set of completed process numbers at the monitoring stage, and $\bar{\varepsilon}(= +\infty)$ is mentioned in Section 2.1. In order to manage the buffers for each project, we decompose vector $\boldsymbol{z}$ to n vectors $z_1, z_1, \cdots, z_n$, which represent n projects by:

$$z_x = \text{diag}(z) \otimes \boldsymbol{l}_{x\alpha}, \tag{32}$$

where $x \in \{1, \cdots, n\}$, and $\boldsymbol{z}$ has the following properties:

$$[z]_i = [\boldsymbol{F}_{\beta\alpha} \otimes v]_i = \begin{cases} t_i : \ i \in \eta, \\ \bar{\varepsilon} : \text{otherwise}, \end{cases} \tag{33}$$

In order to know the projects' performance at the monitoring stage, we calculate the rates of the PBs' consumption and the elapsed times. These values are determined for each project as the following procedure:

- **Ratio of the PBs' consumption (buffer used, %):**
  First, we classify the original PB for each project (x):

$$b_{px} = \boldsymbol{b}_p^T \otimes \boldsymbol{C}_0 \otimes l_{x\alpha}, \tag{34}$$

where $\boldsymbol{b}_p$ is the size of the PBs obtained from Eq. (19). Then, we define the PB consumed by the completed processes for project (x) using the following vector:

$$(\boldsymbol{b}_x)_{used} = \text{diag}(x_{\gamma E}^{+}) \odot z_x, \tag{35}$$

where $\boldsymbol{x}_{\gamma R}^{+}$ presents the earliest completion times of the system after the CCPM application, derived in Section 2.3. Finally, the consumption rate of the PB at the monitoring stage is calculated through vector $\boldsymbol{r}_b$ as:

$$\boldsymbol{r}_{bx} = (\boldsymbol{b}_x)_{used}^{\otimes \frac{1}{b_{px}}}. \tag{36}$$

- **Ratio of the progress time over the critical path duration (time elapsed, %)**
  First, we calculate the duration time of the critical path for each project (x):

$$d_x = (\text{diag}(u'_{\gamma L}) \odot \boldsymbol{y}_{\gamma E})^T \otimes \boldsymbol{C}_0 \boldsymbol{l}_{x\alpha} \tag{37}$$

where $u'_{\gamma L} = \boldsymbol{L}_\alpha^T \odot \boldsymbol{x}_{\gamma L}^{-}$, $u'_{\gamma L}$ expresses the latest feeding times on the critical paths, and $\boldsymbol{x}_{\gamma L}^{-}$ and $\boldsymbol{y}_{\gamma E}$ are the latest starting times and the output times of the system after the CCPM application, derived in Section 2.3. Then, we define the completion time of processes at the monitoring stage without considering the effect of input times as:

$$\boldsymbol{z}'_x = (\boldsymbol{l}_{x\alpha}^T \odot \boldsymbol{B}_0 \boldsymbol{u}_{\gamma L}) \odot \boldsymbol{z}_x. \tag{38}$$

where $\boldsymbol{u}_{\gamma L}$ expresses the latest feeding times which is derived in Section 2.3. Finally, the ratio of the progress time over the estimated completion time is obtained through vector $\boldsymbol{r}_{tx}$ as:

$$\boldsymbol{r}_{tx} = \boldsymbol{z}'^{\otimes \frac{1}{dx}}_x. \tag{39}$$

The ratio of buffer consumption (%) versus the corresponding progress time (%) is plotted, respectively for all projects through the completed processes on the same fever chart (Figure 1). The chart expresses the current status of all projects and their interaction. Thus, managers can make a proper decision for protecting each project and the entire system.

4. NUMERICAL EXAMPLE

A numerical example for a simple system is presented to facilitate better understanding of the proposed framework.

Figure 3 shows a simple system with two projects consisting of three inputs, two outputs and eight processes. Note that the dashed arrow from process (3) to (6) is considered as the precedence constraint between the two projects.

- Project 1 consists of processes (1), (2), (3), and (5).
- Project 2 consists of processes (4), (6), (7), and (8).

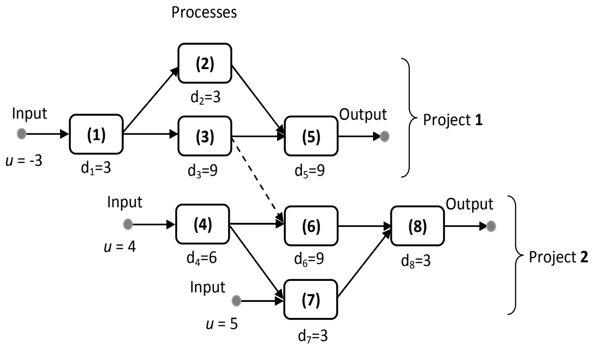

Figure 3. A simple system with two projects (1 and 2).

4.1 Determination of Time Buffers

Using the MPL representation introduced in Section 2.2, the representation matrices for the entire system are given as follows:

$$\boldsymbol{F}_0=\begin{bmatrix}\varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon\\ \boldsymbol{e}&\varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon\\ \boldsymbol{e}&\varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon\\ \varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon\\ \varepsilon&\boldsymbol{e}&\boldsymbol{e}&\varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon\\ \varepsilon&\varepsilon&\boldsymbol{e}&\boldsymbol{e}&\varepsilon&\varepsilon&\varepsilon&\varepsilon\\ \varepsilon&\varepsilon&\varepsilon&\boldsymbol{e}&\varepsilon&\varepsilon&\varepsilon&\varepsilon\\ \varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon&\boldsymbol{e}&\boldsymbol{e}&\varepsilon\end{bmatrix},\ \boldsymbol{C}_0=\begin{bmatrix}\varepsilon&\varepsilon&\varepsilon&\varepsilon&\boldsymbol{e}&\varepsilon&\varepsilon&\varepsilon\\ \varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon&\boldsymbol{e}\end{bmatrix},$$

$$\boldsymbol{P}_k=\mathrm{diag}(3,3,9,6,9,9,3,3)\ \ \boldsymbol{B}_0=\begin{bmatrix}\boldsymbol{e}&\varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon\\ \varepsilon&\varepsilon&\varepsilon&\boldsymbol{e}&\varepsilon&\varepsilon&\varepsilon&\varepsilon\\ \varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon\end{bmatrix}^T.$$

Assuming that the initial condition is $\boldsymbol{x}(0)=\varepsilon_{81}$ and the input times $\boldsymbol{u}=[-3\ 4\ 5]^T$, the output times and the critical path are identified as $\boldsymbol{y}_E=(18\ 22)^T$ and $\alpha=\{1,3,4,5,6,8\}$. respectively.

Moreover, we apply the MPL representation presented in Section 2.3 to the system. First, the processing times are given by $(3,3,9,6,9,9,3,3)^{\otimes 1/3}$.

From Eq. (17), we define two vectors as:

$$\boldsymbol{l}_1=(e\ e\ e\ \varepsilon\ e\ \varepsilon\ \varepsilon\ \varepsilon)^T,\quad \boldsymbol{l}_2=(\varepsilon\ \varepsilon\ \varepsilon\ e\ \varepsilon\ e\ e\ e)^T.$$

The matrix $\boldsymbol{L}$ given in Eq. (18) is:

$$\boldsymbol{L}=\begin{bmatrix}e&e&e&\varepsilon&e&\varepsilon&\varepsilon&\varepsilon\\ \varepsilon&\varepsilon&\varepsilon&e&\varepsilon&e&e&e\end{bmatrix},$$

- **Project buffer**

Since the critical path is identified as $\alpha=\{1,3,4,5,6,8\}$, and $[C_0]_{15}=e$ and $[C_0]_{28}=e$ the positions of the PBs are behind processes (5) and (8). From Eqs. (19) and (20), we obtain the sizes of the PBs as $\boldsymbol{b}_p=(7\ 6)^T$.

- **Feeding buffer**

From Eqs. (21)–(23), we obtain the positions of the FBs inserted at the transitions of process (2) → (5) and process (7) → (8). The sizes of the FBs are determined as $\boldsymbol{b}_f=(\varepsilon\ 1\ \varepsilon\ \varepsilon\ \varepsilon\ \varepsilon\ 1\ \varepsilon)^T$.

- **Capacity buffer**

From Eqs. (24)–(26), a CB is inserted at the transition of process (3) → (6). Moreover, the size of the CB at the transition from project 1 to 2 is $(\boldsymbol{b}_c)_{12}=(\varepsilon\ \varepsilon\ 4\ \varepsilon\ \varepsilon\ \varepsilon\ \varepsilon\ \varepsilon)^T$.

Figure 4 shows the system with the time buffers inserted for the multi-project system in Figure 3.

- **The system with the CCPM applied**

The matrices for representing the structure of the latter system are redefined as follows. Using Eqs. (27)–(29), we obtain the adjacency and output matrices as:

$$\boldsymbol{F}_0=\begin{bmatrix}\varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon\\ \boldsymbol{e}&\varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon\\ \boldsymbol{e}&\varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon\\ \varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon\\ \varepsilon&1&\boldsymbol{e}&\varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon\\ \varepsilon&\varepsilon&4&\boldsymbol{e}&\varepsilon&\varepsilon&\varepsilon&\varepsilon\\ \varepsilon&\varepsilon&\varepsilon&\boldsymbol{e}&\varepsilon&\varepsilon&\varepsilon&\varepsilon\\ \varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon&\boldsymbol{e}&1&\varepsilon\end{bmatrix},\ \boldsymbol{C}_\gamma=\begin{bmatrix}\varepsilon&\varepsilon&\varepsilon&\varepsilon&7&\varepsilon&\varepsilon&\varepsilon\\ \varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon&6\end{bmatrix},$$

$$\boldsymbol{P}_k=\mathrm{diag}(1,1,3,2,3,3,1,1)\ \ \boldsymbol{B}_0=\begin{bmatrix}\boldsymbol{e}&\varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon\\ \varepsilon&\varepsilon&\varepsilon&\boldsymbol{e}&\varepsilon&\varepsilon&\varepsilon&\varepsilon\\ \varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon&\varepsilon\end{bmatrix}^T$$

Using Eqs. (10)-(16), the earliest completion times and the corresponding output times are calculated as $x_{\gamma E}^{+}=(-2\ -1 1\ 6\ 4\ 9\ 7\ 10)^T$ and $\boldsymbol{y}_{\gamma E}=(11\ 16)$ respectively. The latest feeding times are obtained as $\boldsymbol{u}_{\gamma L}=(-3\ 4\ 7)^T$. Moreover, the critical path is identified as $\alpha=\{1,3,4,5,6,8\}$.

4.2 Buffer Management

After the insertion of the time buffers, the multi-project system is monitored and controlled through the

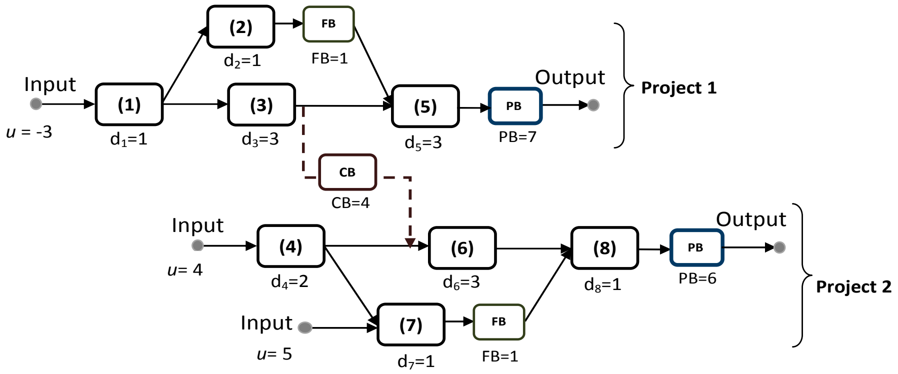

Figure 4. The multi-project system after the application of critical chain project management method.

buffer management proposed in Section 3.2. From Eq. (30), we obtain two vectors as:

$$\boldsymbol{l}_{1\alpha} = [e\ \varepsilon\ e\ \varepsilon\ e\ \varepsilon\ \varepsilon\ \varepsilon]^T,$$
$$\boldsymbol{l}_{2\alpha} = [\varepsilon\ \varepsilon\ \varepsilon\ e\ \varepsilon\ e\ \varepsilon\ e]^T.$$

We assume here that the set of the completed processes η and the vector of the actual completion times $\boldsymbol{z}$ have been collected at the motoring stage as:

$$\eta = \{1, 2, 3, 4, 6, 7\}$$
$$\boldsymbol{z} = (0\ 2\ 6\ 7\ \overline{\varepsilon}\ 11\ 8\ \overline{\varepsilon})^T$$

Using Eq. (32), we decompose $\boldsymbol{z}$ into:

$$\boldsymbol{z}_1 = \text{diag}(\boldsymbol{z}) \otimes \boldsymbol{l}_{1\alpha} = [0\ \varepsilon\ 6\ \varepsilon\ \overline{\varepsilon}\ \varepsilon\ \varepsilon\ \varepsilon]^T$$
$$\boldsymbol{z}_2 = \text{diag}(\boldsymbol{z}) \otimes \boldsymbol{l}_{2\alpha} = [\varepsilon\ \varepsilon\ \varepsilon\ 7\ \varepsilon\ 11\ \varepsilon\ \overline{\varepsilon}]^T$$

- **Ratio of the PB's consumption**
  - Classify the original PB of each project:
  $$b_{p1} = b_p^T \otimes \boldsymbol{C}_0 \otimes \boldsymbol{l}_{1\alpha} = 7$$
  $$b_{p2} = b_p^T \otimes \boldsymbol{C}_0 \otimes \boldsymbol{l}_{2\alpha} = 6$$
  - Define the PBs consumed by the completed processes on each project (buffer used):
  $$(\boldsymbol{b}_1)_{used} = \text{diag}(x_{\gamma E}^{+}) \odot \boldsymbol{z}_1$$
  $$= [2\ \varepsilon\ 5\ \varepsilon\ \overline{\varepsilon}\ \varepsilon\ \varepsilon\ \varepsilon]^T,$$
  $$(\boldsymbol{b}_2)_{used} = \text{diag}(x_{\gamma E}^{+}) \odot \boldsymbol{z}_2$$
  $$= [\varepsilon\ \varepsilon\ \varepsilon\ 1\ \varepsilon\ 2\ \varepsilon\ \overline{\varepsilon}]^T,$$

- **Ratio of the corresponding progress time over the expected duration time**
  - Calculate the duration time of the critical path for each project:
  $$\boldsymbol{u}'_{rL} = \boldsymbol{L}_{\alpha}^T \odot \boldsymbol{x}_{\gamma L}^{-} = \begin{bmatrix} -3 \\ 4 \end{bmatrix}.$$
  $$d_1 = (\text{diag}(u'_{\gamma L}) \odot \boldsymbol{y}_{rE})^T \otimes \boldsymbol{C}_0 \boldsymbol{l}_{1\alpha} = 14$$
  $$d_2 = (\text{diag}(u'_{\gamma L}) \odot \boldsymbol{y}_{rE})^T \otimes \boldsymbol{C}_0 \boldsymbol{l}_{2\alpha} = 12$$
  - Define the progress time at the monitoring stage without considering the effect of the input time (time used):
  $$\boldsymbol{z}'_1 = (\boldsymbol{l}_{1\alpha}^T \odot \boldsymbol{B}_0 \boldsymbol{u}_{\gamma L}) \odot \boldsymbol{z}_1 = [3\ \varepsilon\ 9\ \varepsilon\ \overline{\varepsilon}\ \varepsilon\ \varepsilon]^T,$$
  $$\boldsymbol{z}'_2 = (\boldsymbol{l}_{2\alpha}^T \odot \boldsymbol{B}_0 \boldsymbol{u}_{\gamma L}) \odot \boldsymbol{z}_2 = [\varepsilon\ \varepsilon\ \varepsilon\ 3\ \varepsilon\ 7\ \varepsilon\ \overline{\varepsilon}]^T,$$
  - Obtain the ratio of the progress time over the duration time of the critical path (time used, %):
  $$\boldsymbol{r}_{t1} = z_1'^{\otimes \frac{1}{d_1}} = \left[\frac{3}{14}\ \varepsilon\ \frac{9}{14}\ \varepsilon\ \overline{\varepsilon}\ \varepsilon\ \ \varepsilon\ \varepsilon\right]^T,$$
  $$\boldsymbol{r}_{t1} = z_2'^{\otimes \frac{1}{d_2}} = \left[\varepsilon\ \varepsilon\ \varepsilon\ \frac{3}{12}\ \varepsilon\ \frac{7}{12}\ \varepsilon\ \overline{\varepsilon}\right]^T.$$

The ratio of buffer consumption versus the corresponding progress time is plotted on a fever chart as shown in Figure 5. The plotted points express the statuses of the completed processes along the critical paths. The chart implies the following:

- The performance of project 1 is expressed by processes (1) and (3). In particular, process (3) shows the latest status which belongs to the red zone. It means that project 1 will possibly be late and managers should initiate the extra action to avoid further buffer consumption.
- The performance of project 2 is expressed by processes (4) and (6). Process (6) shows the latest status which belongs to the green zone. It means that project 2 is going well and need not take an action.
- Since a capacity buffer was inserted between projects 1 and 2 at the transition of process (3) → (6), though process (3) takes a longer time to complete which leads project 1 to be at risk, project 2 is still all right. Thus, it signifies that managers need to take the actions which should focus on project 1 only.

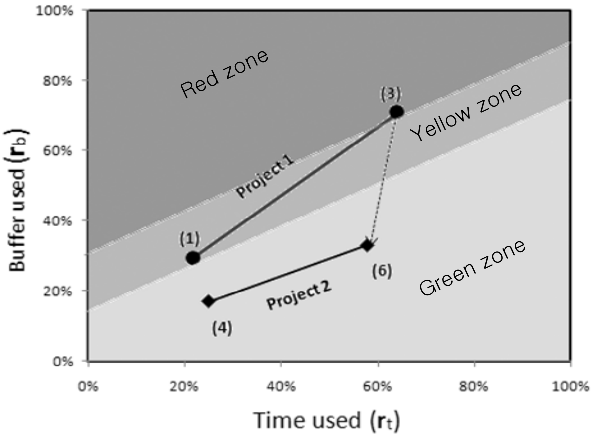

Figure 5. Fever chart for the multi-project system in Figure 4.

5. CONCLUSION

We have proposed an MPL representation for describing a buffer management in the CCPM method focusing on multi-project systems. By introducing an inspected vector which represents the actual completion time of the completed processes at the monitoring stage, the managers can survey the performance of all projects frequently. Moreover, the proposed method can analyze the complex system to estimate the rates of the consumed project buffers and the elapsed progress times readily for expressing the current status of all projects and their interaction, which support managers to easily monitor each project, effectively managing the entire system.

Since we have developed a deterministic model for inserting and controlling the time buffers, for practical cases, a stochastic model, which takes into account the analysis of the external uncertainties, will be an important future work.

ACKNOWLEDGMENTS

Yoshinori Takei has been supported in part by a research grant from JSPS Grants-in-Aid for Scientific Research No. 23650005. Hiroyuki Goto has been supported in part by a research grant from JSPS Grants-in-Aid for Scientific Research No. 23510163. Hirotaka Takahashi has used the facilities of Earthquake Research Institute (ERI), The University of Tokyo. He has also been supported in part by the JSPS Grant-in-Aid for Scientific Research No. 23740207.

REFERENCES

Cohen, I., Mandelbaum, A., and Shtub, A. (2004), Multi-project Scheduling and Control: A process-based comparative study of the Critical Chain Methodology and Some Alternatives, *Project Management Journal*, **35**(2), 39-50.

Goldratt, E. M. (1990), *Theory of Constraints*, North River Corp., Great Barington, MA.

Goto, H. (2010), Modelling methods based on discrete algebraic systems. In: Goti, A. (ed.), *Discrete Event Simulations*, Sciyo, Rijeka, Croatia, 35-62.

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

Heidergott, B., Jan Olsder, G., and van der Woude, J. W. (2006), *Max Plus at Work: Modeling and Analysis of Synchronized Systems*, Princeton University Press, Princeton, NJ.

Herroelen, W., Leus, R., and Demeulemeester, E. (2002), Critical chain project scheduling: do not oversimplify, *Project Management Journal*, **33**, 46-60.

Kasahara, M., Takahashi, H., and Goto, H. (2009), On a buffer management policy for CCPM-MPL representation, *International Journal of Computational Science*, **3**(6), 593-606.

Leach, L. P. (2005), *Critical Chain Project Management*, Artech House, Boston, MA.

Takahashi, H., Goto, H., and Kasahara, M. (2009), Toward the application of a critical-chain-project-management-based framework on max-plus linear systems, *Industrial Engineering and Management Systems*, **8**(3), 155-161.

Truc, N. T. N., Goto, H., Takahashi, H., and Yoshida, S. (2011), Enhanced framework for determining time buffers in the MPL-CCPM representation, *International Journal of Information Science and Computer Mathematics*, **4**(1), 1-18.

Truc, N. T. N., Goto, H., Takahashi, H., Yoshida, S., and Takei, Y. (2012), Critical chain project management based on a max-plus linear representation for determining time buffers in multiple projects, *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, **6**(5), 715-727.

Tukel, O. I., Rom, W. O., and Eksioglu, S. D. (2006), An investigation of buffer sizing techniques in critical chain scheduling, *European Journal of Operational Research*, **172**(2), 401-416.

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

Yoshida, S., Takahashi, H., and Goto, H. (2011), Resolution of time and worker conflicts for a single project in a max-plus linear representation, *Industrial Engineering and Management Systems*, **10**(4), 279-287.