Engineering Applications of Artificial Intelligence 22 (2009) 603–607

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Brief Paper

Robust MPL scheduling considering the number of in-process jobs

Hiroyuki Goto *

Department of Management and Information Systems Science, Nagaoka University of Technology, 1603-1 Kamitomiokamachi, Nagaoka, Niigata 940-2188, Japan

ARTICLE INFO

Article history:
Received 16 March 2007
Received in revised form
15 November 2007
Accepted 17 November 2008
Available online 20 January 2009

Keywords:
Max-plus-linear systems
State-space representation
Discrete-time system robustness
Process disturbances
Rescheduling

ABSTRACT

A robust scheduling method for max-plus linear systems is proposed. A principal concern in scheduling problems is how to accomplish robustness against external disturbances. To accomplish this, methods based on model predictive control (MPC) have been put forward to control system parameters or control inputs. In this context, we recently proposed a method for indirectly controlling the state variables by utilizing the idea of dead time. The idea imposes constraints for upper bounds of in-processing jobs between facilities, whereas several practical systems also consider the lower bounds. This paper, therefore, considers a modeling and robust scheduling method that takes into account both constraints. A numerical simulation for a transportation system is also presented in order of the method's effectiveness.

1. Introduction

This paper derives a max-plus linear (MPL) state-space representation (hereafter called MPL representation) for repetitive systems with multiple inputs and multiple outputs (MIMO) structure taking into account the number of in-processing jobs, and proposes a robust scheduling method for corresponding systems. Methods for modeling discrete event systems utilizing MPL representation are applied to fields of production systems (De Schutter and Boom, 2001), chemical plants (Schullerus and Krebs, 2001), and transportation systems (Braker, 1991; Heidergott and De Veries, 2001; De Schutter and Boom, 2002). Much attention has been paid to these applications.

In MPL representation, changes of states in systems can be described by linear equations of max-plus algebra (Cohen et al., 1989; Baccelli et al., 1992; Heidergott et al., 2006). These equations are similar to state-space representation in modern control theory. The simplest form is stated by three constant matrices; the system, input, and output matrices. It can describe systems in which there are features such as; (1) no-concurrency in identical resource, (2) parallel execution of multiple tasks and/or (3) synchronization of multiple tasks. Based on these features, extensions whose fundamental idea is similar to queue (Krivulin, 1996) or dead time (Goto, 2008) can consider the maximum number of in-processing jobs that can exist in a single facility or between facilities. By this extension, the method can also be applied to scheduling problems for productions systems with limited storage space or for train traffic control systems. Note that the terms of 'facility', 'job', and 'task' are interpreted as 'machine', 'batch' and 'processing' in production systems, respectively. Moreover, transportation systems interpret them as 'station', 'train', and 'transportation', respectively.

A principal concern in scheduling problems is the robustness of the solution. For instance, when the occupation times of facilities and/or the predicted output time are changed, the state variables, input variables, and the output ones desirably keep their values close to their previous ones. Methods based on model predictive control (De Schutter and Boom, 2001; Boom and De Schutter, 2002) are often utilized to conform to this requirement by essentially directly controlling the input variables or system parameters. Other methods based on queue or dead time control the state variables indirectly by giving constraints regarding the upper limits of in-process jobs. However, they may often be useful for robustness when setting constraints for the lower limits of in-process jobs. For example, train dispatchers sometimes instruct to retard some trains to avoid congestion, when another train has been delayed. Moreover, the TOC theory (Goldratt, 1990; Stein, 1996) which has garnered much attention recently, requires that a requisite minimum inventory in previous steps of the bottleneck processes and the shipping ones be stored. It also requires that the non-bottleneck processes be subordinated to the bottleneck ones.

Hence, this paper extends the conventional MPL representation and proposes a robust scheduling method taking into account the upper and lower limits of in-process jobs. The extended state-space representation is a set of simultaneous linear equations in max-plus algebra, and its solution provides a robust schedule. In the latter part of this paper, a simple numerical simulation for a

* Tel./fax: +81 258 47 0322.
E-mail address: hgoto@kjs.nagaokaut.ac.jp

0952-1976/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.engappai.2008.11.007

transportation system is carried out, confirming the effectiveness of the proposed method.

2. Extended MPL representation

In this section, first, mathematical preliminaries are given and then an extended MPL representation that can take into account the minimum number of in-process jobs derived.

2.1. Mathematical preliminaries

Denote the real field by $\boldsymbol{R}$, in $\mathbf{D}=\boldsymbol{R}\cup\{-\infty\}$ and for $x, y\in\mathbf{D}$, the following operators are defined.

$$x\oplus y=\max(x,y),\quad x\otimes y=x+y$$

The operator for multiplication $\otimes$ is abbreviated as in conventional algebra when no confusion is likely to arise. Unit elements for $\oplus$ and $\otimes$ are denoted by $\varepsilon(=-\infty)$ and $e(=0)$, respectively. If $m\leqslant n$

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

For a matrix $\boldsymbol{X}\in\mathbf{D}^{m\times n}$, $[\boldsymbol{X}]_{ij}$ represents the (i,j)-th element of $\boldsymbol{X}$. If $\boldsymbol{X}\in\mathbf{D}^{m\times l}$, $\boldsymbol{Y}\in\mathbf{D}^{l\times p}$,

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

Regarding the unit matrix, $\boldsymbol{\varepsilon}_{mn}\in\mathbf{D}^{m\times n}$ represents the unit matrix for $\oplus$ whose elements are all ε, and $\boldsymbol{e}_m\in\mathbf{D}^{m\times m}$ for $\otimes$ in which only diagonal elements are e and all off-diagonal elements are ε.

2.2. Formulation

Suppose the following constraints are imposed on systems.

- The number of facilities is n, p is the number of external inputs and q the external outputs.
- Constraints on in-process jobs are imposed only between facilities, and not with any external inputs or outputs.
- Other constraints such as no-concurrency, parallel processing, or synchronization are taken into account in an analogous way to conventional MPL representation.

For simplicity, let us ignore transportation times between facilities. Fig. 1 shows relevant constraints regarding facility i $(1\leqslant i\leqslant n)$. Denote the occupation time by $d_i(\geqslant 0)$, the job number by k, and the earliest completion time by $[\boldsymbol{x}_E(k)_i]$, where the suffix E represents the earliest time. $[\boldsymbol{u}(k)]_j$ represents the feeding time for external input j. $\mathbf{R}_i$, $\mathbf{P}_i$, $\mathbf{M}_{ih}$, and $\mathbf{K}_{il}$ represent a collection of numbers of the preceding facilities, those of the external inputs and the downstream facilities in which the maximum number of in-process jobs with facility i is h, and those of upstream facilities in which the minimum number of in-process jobs is l. Note that the constraints regarding the in-process jobs are imposed on the next two time instants; the starting point in the i-th facility and the completion point in other facilities. Other events can be adopted for this formulation, but are more complicated. The following matrices $\boldsymbol{P}$, $\boldsymbol{F}_0$, $\boldsymbol{H}_0^{(h)}$, $\boldsymbol{L}_0^{(l)}\in\mathbf{D}^{n\times n}$, $\boldsymbol{B}_0\in\mathbf{D}^{n\times p}$, $\boldsymbol{C}_0\in\mathbf{D}^{q\times n}$ are introduced to specify the structure of the system.

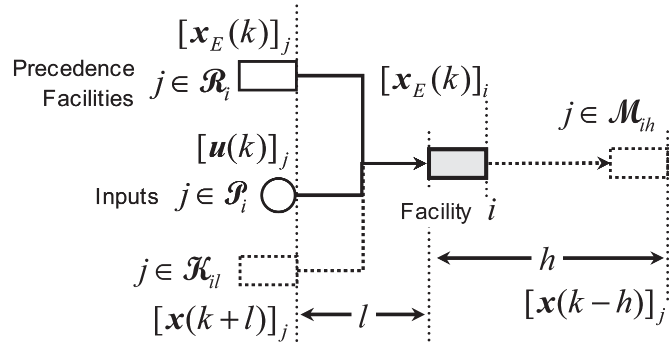

Fig. 1. Relevant facilities and external inputs regarding to the i-th facility.

$$[\boldsymbol{P}]_{ij}=\begin{cases} d_i & :\text{if } i=j\\ \varepsilon & :\text{if } i\neq j\end{cases}$$

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

$$[\boldsymbol{H}_0^{(h)}]_{ij}=\begin{cases} e & :\text{the maximum number of in-process jobs is } h \text{ between facility } i \text{ and its downstream facility } i\\ \varepsilon & :\text{there are not any constraints for in-process jobs between facilities } i \text{ and } j\end{cases}$$

$$[\boldsymbol{L}_0^{(l)}]_{ij}=\begin{cases} e & :\text{the minimum number of in-process jobs is } l \text{ between facility and its upstream facility } j\\ \varepsilon & :\text{there are not any constraints for in-process jobs between facilities } i \text{ and } j\end{cases}$$

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

$$[\boldsymbol{C}_0]_{ij}=\begin{cases} e & :\text{external output } i \text{ has a preceding facility } j\\ \varepsilon & :\text{external output } i \text{ does not have a preceding facility } j\end{cases}$$

In accord with the discussions in (Goto and Masuda, 2006), these constraints in Fig. 1 can be formulated as follows.

$$\begin{aligned}
&[\boldsymbol{x}_E(k)]_i-d_i\\
&=\bigoplus_{j\in\mathbf{R}_i}[\boldsymbol{x}_E(k)]_j\oplus\bigoplus_{j\in\mathbf{P}_i}[\boldsymbol{u}(k)]_j\oplus\bigoplus_{h=1}^{H}\bigoplus_{j\in\mathbf{M}_{ih}}[\boldsymbol{x}(k-h)]_j\oplus\bigoplus_{l=1}^{L}\bigoplus_{j\in\mathbf{K}_{il}}[\boldsymbol{x}(k+l)]_j\\
&=\bigoplus_{j=1}^{n}([\boldsymbol{F}_0]_{ij}+[\boldsymbol{x}(k)]_j)\oplus\bigoplus_{j=1}^{p}([\boldsymbol{B}_0]_{ij}+[\boldsymbol{u}(k)]_j)\oplus\bigoplus_{h=1}^{H}\bigoplus_{j=1}^{n}([\boldsymbol{H}_0^{(h)}]_{ij}+[\boldsymbol{x}(k-h)]_j)\\
&\quad\oplus\bigoplus_{l=1}^{L}\bigoplus_{j=1}^{n}([\boldsymbol{L}_0^{(l)}]_{ij}+[\boldsymbol{x}(k+l)]_j)\\
&=[\boldsymbol{F}_0\boldsymbol{x}_E(k)]_i\oplus[\boldsymbol{B}_0\boldsymbol{u}(k)]_i\oplus\bigoplus_{h=1}^{H}[\boldsymbol{H}_0^{(h)}\boldsymbol{x}(k-h)]_i\oplus\bigoplus_{l=1}^{L}[\boldsymbol{L}_0^{(l)}\boldsymbol{x}(k+l)]_i
\end{aligned}\tag{1}$$

where $\boldsymbol{H}$ and $\boldsymbol{L}$ represent the maximal of the maximum number of in-process jobs and those for the minimum number in the entire system, respectively. By transposing d_i on the left size of Eq. (1) to the right side, the next equation is obtained using $\boldsymbol{P}$.

$$[\boldsymbol{x}_E(k)]_i=\left[\boldsymbol{P}[\boldsymbol{F}_0\boldsymbol{x}_E(k)\oplus\boldsymbol{B}_0\boldsymbol{u}(k)\oplus\bigoplus_{h=1}^{H}\boldsymbol{H}_0^{(h)}\boldsymbol{x}(k-h)\oplus\bigoplus_{l=1}^{L}\boldsymbol{L}_0^{(l)}\boldsymbol{x}(k+l)]\right]_i\tag{2}$$

This holds true for all i $(1\leqslant i\leqslant n)$. Here the latest completion times for jobs (k)–$(k+N-1)$ are considered. Eq. (2) holds true even