Proceedings of the
46th IEEE Conference on Decision and Control
New Orleans, LA, USA, Dec. 12-14, 2007

Monotonically Improving Error Bounds for a Sequence of Approximations for Makespan Minimization of Batch Manufacturing Systems

W. Weyerman, S. Warnick

Information and Decision Algorithms Laboratories
Department of Computer Science, Brigham Young University, Provo, UT 84602
http://idealabs.byu.edu

***Abstract*— We consider a batch manufacturing system model with no intermediate storage and a single machine at each workstation. These models have applications in fields such as chemical processing, computer systems, and manufacturing. We develop a max-plus representation for this system with linear dynamics and show relevant properties of this system. In particular, we express a sequence of increasingly complex approximations to the minimum makespan problem in the max-plus algebra. Using this formulation, we compute monotonically improving error bounds for these approximations, guaranteeing a decrease in error as more computational effort is used in the approximation.**

I. INTRODUCTION

Multipurpose batch manufacturing systems are used to process a variety of job types using a fixed set of resources, commonly referred to as machines. The batch property allows several jobs of a given type to be processed simultaneously by a single machine. This complicates the well known job-shop problem in that different machines may process batches of different sizes. Thus, one machine may have to finish processing 3 different batches before the next machine in the job type's route can begin processing one batch. The job shop problem assumes the capacity, or batch size, of each machine is equal; this is clearly a special case. The batch manufacturing system model is useful for a wide range of applications including manufacturing systems; chemical processing plants; computing resources, including mutual exclusion problems, processor resources, and parallel architectures; and services.

Several papers have been published regarding various aspects of the batch scheduling problem. As with the job shop, some have approached the problem of scheduling small systems of one or two machines [2], [4], [8], [10]. The problem of how to group jobs to form batches is treated in [3], [5], [6]. A graphical solution to the general multipurpose batch plant is given in [12] which works well for simple examples. However, when there are machines of drastically different sizes or complex recipes, the problem grows to an intractable size. Others such as [9] and [11] use a graphical representation of a multipurpose batch plant to derive a mixed integer linear program (MILP) formulation to determine the exact answer to solve several objectives. Because these methods are exact, they are also computationally intractable for a complex manufacturing system. Two heuristic methods of solution to the minimum makespan problem are given in [1], the better of the two reduces decision variables in the MILP by a linear factor. A linear reduction in decision variables in an MILP is not very helpful because of the complexity of solving an MILP; the problem still rapidly grows to an intractable size.

In this paper we analyze the heuristic given in [14] on a batch manufacturing system with single machine workstations by developing a max-plus representation for these systems. Using this representation we show that the error incurred when using the heuristic is bounded and we compute these bounds. We further show that as the complexity of the approximation increases, the error bound decreases. We conclude with an example.

II. MAX-PLUS ALGEBRA

We will briefly discuss the max-plus algebra as it is presented in [7]. The max-plus algebra is defined over $\mathbb{R}_{max} = \mathbb{R} \cup -\infty$. We will define three operations for scalars:

$$\begin{aligned} &\forall a, b \in \mathbb{R}_{max} \\ &\quad a \oplus b = \max(a, b) \\ &\quad a \otimes b = a + b \\ &\quad a \oslash b = a - b. \end{aligned}$$

The zero element is defined as $\epsilon = -\infty$, and the unit element is defined as $e = 0$.

Matrix arithmetic is also defined. For matrices, $A, B \in \mathbb{R}_{max}^{n\times l}$, $C \in \mathbb{R}_{max}^{l\times m}$ these are defined as:

$$\begin{aligned} [A \oplus B]_{ij} &= a_{ij} \oplus b_{ij} \\ [B \otimes C]_{ik} &= \bigoplus_{j=1}^{l} b_{ij} \otimes c_{jk}. \end{aligned}$$

This work has been supported by generous support from ATK Thiokol, Sandia National Laboratories, and the Office of Research and Creative Activities at BYU. Please direct questions and comments to W. Sam Weyerman at wsweyerman@gmail.com or Sean Warnick at sean@cs.byu.edu.

The zero vector and the unit vector are given by

$$\boldsymbol{\epsilon} = \begin{bmatrix} \epsilon \\ \vdots \\ \epsilon \end{bmatrix}$$

$$\mathbf{e} = \begin{bmatrix} e \\ \vdots \\ e \end{bmatrix}.$$

The identity matrix is

$$I = \begin{bmatrix} e & \epsilon & \dots & \epsilon \\ \epsilon & e & \dots & \epsilon \\ \vdots & & & \\ \epsilon & \epsilon & \dots & e \end{bmatrix}.$$

We can now define a linear state-space system in the max-plus algebra. For $\mathbf{x}(k) \in \mathbb{R}^n_{max}$, and $A \in \mathbb{R}^{n\times n}_{max}$, there is a linear autonomous system,

$$\mathbf{x}(k+1) = A \otimes \mathbf{x}(k).$$

Definition 2.1: We say a max-plus autonomous system is stable if

$$\forall i \exists v \in \mathbb{R} \ \lim_{k\to\infty} x_i(k) \oslash x_1(k) = v.$$

Throughout this paper we will use the convention that for any $y \in \mathbb{R}_{max}$, the indeterminate form $y \oplus (\epsilon \oslash \epsilon) = y$.

We will also define the 1-norm in the max-plus algebra.

Definition 2.2: The 1-norm of a max-plus vector, $b \in \mathbb{R}^n_{max}$ is

$$||b||_1 = \bigoplus_{i=1}^{n} b_i = \mathbf{e}^T \otimes b$$

This norm induces a norm on a matrix.

Definition 2.3: The 1-induced norm of an operator $A \in \mathbb{R}^{n\times m}_{max}$ is

$$||A||_1 = max_x(||A \otimes x||_1 \oslash ||x||_1).$$

Theorem 2.1: Given a matrix $A \in \mathbb{R}^{n\times m}_{max}$, the max-plus 1-induced norm of A is

$$||A||_1 = \max_{ij} a_{ij}.$$

Proof supplied in [13].

Theorem 2.2: Given a matrix $A \in \mathbb{R}^{n\times m}_{max}$,

$$\min_x ||A \otimes x||_1 \oslash ||x||_1 = \min_i [\mathbf{e}^T \otimes A]_i.$$

Proof supplied in [13].

III. System and Problem Formulation

We consider a batch manufacturing system model with no intermediate storage. We have a factory F composed of m workstations W. So $F = \{W_1, W_2, \dots, W_m\}$. It processes a suite of n job types given by the enumeration $\mathcal{J} = \{1, 2, \dots, n\}$. Job type j is characterized by a recipe $(\mathbf{c},\boldsymbol{\tau})(j)$. Where $c_i(j)$ represents the capacity of workstation i when processing job type j, and $\tau_i(j)$ represents the run time of workstation i when processing job type j. Note that this representation requires each job type to follow the same route through the factory, namely the route that starts at workstation 1, then proceeds to workstation 2, and so on until workstation m.

In order to limit the decision making to the sequence of job types entering the factory we make some assumptions about the factory. We assume permutation schedules: each machine follows the same sequence of job types. We also assume that no workstation will begin operation of a batch until the machine is completely filled. Therefore, each time a job type is sequenced, the number of jobs that are sequenced is equal to the least common multiple of the capacities of each workstation for that job type, this is called a load and is denoted $L(j)$. This also requires that each workstation process a set number of batches each time a job type is scheduled, workstation i will process $B_i(j) = L(j)/c_i(j)$ batches. We employ a no-idling policy: a workstation is utilized as soon as it is available and there are jobs waiting for it.

The step index of our system, $k \in \mathbb{N}$, specifies where in the sequence we are. Thus, each job type that is scheduled represents a single step. We define the state of the system, $x(k) \in \mathbb{R}^m$, to represent the time that each machine is available for processing after the k^{th} step. The input to the system, $u(k) \in \mathcal{J}$, specifies the job type to be processed at step k.

This system can be represented as some discrete time dynamical system,

$$x(k+1) = f(x, u, k); \quad y(k) = g(x, u, k).$$

We define $f(x, u, k)$ using a recursive method based on the precedence structure imposed by our system.

$$\beta_{i,j}(k) = \begin{cases} x_1(k) & i = j = 1, \\ E_{1,j-1}(k) & i = 1, j \neq 1, \\ \max\Big(\beta_{i-1,\lceil \frac{jc_i(u_k)}{c_{i-1}(u_k)}\rceil}(k) & \\ +\tau_{i-1}(u_k), E_{i,j-1}(k)\Big) & otherwise \end{cases} \tag{1}$$

$$E_{i,j}(k) = \begin{cases} 0 & j < 0, or\ i > n, \\ x_i(k) & j = 0 \\ \max\big(\beta_{i,j}(k) + \tau_i(u_k), & \\ E_{i+1,\lceil \frac{jc_i(u_k)}{c_{i+1}(u_k)}\rceil -1}(k)\Big) & otherwise. \end{cases} \tag{2}$$

$$x_i(k+1) = E_{i,B_i}(k). \tag{3}$$

Here $\beta_{i,j}(k)$ refers to the time that workstation i begins processing of batch j of the job type specified by u_k, and $E_{i,j}(k)$ refers to the time that workstation i has completely unloaded batch j. Note that relaxing the no-idling constraint changes the equalities in these equations to greater than or equal to.

We define $g(x, u, k)$ in conventional algebra as

$$g(x, u, k) = ||x(k)||_\infty - ||x(k-1)||_\infty.$$

This is simply the amount that the max value of x is increased by choosing u_k to be the next element in the sequence.

Our objective is to minimize the makespan while reaching a certain quota, $\mathbf{q}$, to get a measure of capacity of the factory with respect to a certain quota. The makespan of a sequence $U = (u_1, u_2, \ldots, u_p)$ of length p is

$$\sum_{k=1}^{p} y(k).$$

We say that a sequence is admissible if for each i, i is in U q_i times.

Thus, the problem that we wish to solve is: given a factory F, a set of job types J, with their associated recipes $(\mathbf{c}, \boldsymbol{\tau})$, and a quota $\mathbf{q}$, find an admissible sequence U to

$$\begin{array}{ll} \min & \sum_{k=1}^{||q||_1} y(k) \\ \text{subject to} & x(k) = f(x,u,k) \\ & y(k) = ||x(k)||_\infty - ||x(k-1)||_\infty . \end{array} \tag{4}$$

IV. A Max-Plus Representation

Because this problem is NP-complete, we wish to analyze this system in order to approximate this problem. However, due to the recursive nonlinear definition of $f(x, u, k)$ we seek a different representation of this system. It turns out that this system has linear dynamics in the max-plus algebra, so we will derive a max-plus representation of the system.

To do this we will consider the system as a heap model. We will represent the time that workstation i spends processing a batch of job type j as a piece of width one and height $\tau_i(j)$. The piece for job type j and workstation i is given by a matrix $P_i(j)$ which is equal to I except that $[P_i(j)]_{ii} = \tau_i(j)$. We represent the precedence of workstation i over $i+1$ using a piece of width two and height 0. This matrix, R_i, is equal to I except that $[R_i]_{i,i+1} = [R_i]_{i+1,i} = e$.

Using these pieces we can construct the heap created by job type j using the algorithm given in [13]. This algorithm multiplies several piece matrices together to arrive at a matrix which we refer to as $A(j)$. We will show that this matrix defines the linear max-plus model for (3). Thus, the system can now be represented in the max-plus algebra as

$$\begin{array}{rcl} x(k+1) & = & A(u_k) \otimes x(k) \\ y(k) & = & ||x(k)||_1 \oslash ||x(k-1)||_1 . \end{array} \tag{5}$$

The 1-norm specified here is the max-plus 1-norm.

A. Properties

To ease notation we will introduce some new symbols. Given A, we define

$$\begin{array}{rl} \xi_{ij} = & a_{ij} - a_{i,j+1}, \\ \delta_i = & a_{i+1,1} - a_{i1}. \end{array}$$

We will give a definition of matrix structure.

Definition 4.1: We will say that a matrix A has the *monotone property* if:

1) $a_{ij} \le a_{i+1,j}$,
2) $a_{ij} \ge a_{i,j+1}$,
3) $\xi_{1i} \ge \xi_{2i} \ge \ldots \ge \xi_{ni} \ge 0,\ \forall i \le n$,
4) $a_{ij} > -\infty,\ \forall i \le j \le n$.

Lemma 4.1: A matrix $B \in \mathbb{R}_{max}^{n\times n}$ with the monotone property maintains Property 2 of the monotone property after finitely many left multiplications of P_i and R_i for some job type with n stages.

Proof supplied in [13].

Lemma 4.2: A matrix $B \in \mathbb{R}_{max}^{n\times n}$ with the monotone property maintains Property 3 of the monotone property after finitely many left multiplications of P_i and R_i for some job type with n stages.

Proof supplied in [13].

Theorem 4.1: Given $(\mathbf{c}, \boldsymbol{\tau})(\alpha)$, $A(\alpha)$ has the monotone property.

This proof is provided in [13].

The fact that A has the monotone property gives two easy results

Corollary 4.1: Given $(\mathbf{c}, \boldsymbol{\tau})(\alpha)$, we have the following:

$$\begin{array}{rcl} ||A(\alpha)||_1 & = & a_{n1}(\alpha) \\ \min_x ||A(\alpha)\otimes x||_1 \oslash ||x||_1 & = & a_{nn}(\alpha). \end{array}$$

Proof: Let A be given. Due to monotonicity of A,

$$\begin{array}{rcl} ||A||_1 & = & \max_{ij} a_{ij} \\ & = & a_{n1}. \end{array}$$

And

$$\begin{array}{rcl} \min_x ||A\otimes x||_1 \oslash ||x||_1 & = & \min_i(\mathbf{e}^T \otimes A) \\ & = & \min_i([a_{n1}, \ldots, a_{nn}]) \\ & = & a_{nn}. \end{array}$$

■

In [13] these systems are shown to be stable in the sense of Definition 2.1.

V. Approximation Method and Bounds

We will express the approximation method given in [14] in the max-plus formulation. For the t-step approximation, given a sequence $U = (u_k, \ldots, u_{k+t})$ we approximate (5) as

$$\begin{array}{rl} \tilde{x}_t(k+1) & = A(u_{k+t}) \otimes \ldots \otimes A(u_k) \otimes x^* \\ \tilde{y}_t(k) & = ||\tilde{x}_t(k)||_1 \oslash ||\tilde{x}_{t-1}(k-1)||_1 . \end{array}$$

Where we define $x^* = \mathbf{e}$. This approximation eases the solution to (4) as it exponentially reduces the number of possible states and therefore number of values to compute at each stage of the dynamic program that solves the problem. This replaces a large NP-complete problem with a much smaller NP-complete problem. Using this approximation, much more complex problems can be solved in practice. These ideas are made precise in [13].

The error of a single step of the t-step approximation is given by

$$\varepsilon_t(k) = |y(k) \oslash \tilde{y}_t(k)|.$$

We can explicitly calculate the error bound for the 0-step approximation for job type j.

Theorem 5.1: The maximum error of the 0-step approximation for job type j is

$$0 \le \varepsilon_0(k) \le \gamma_0(j) = a_{n1}(j) \oslash a_{nn}(j)$$

Proof: Let $A(j)$ be given. The maximum error of the 0-step approximation is given by

$$\gamma_0(j) = \max_x\{(||A \otimes x^*||_1 \oslash ||x*||_1) \oslash (||A \otimes x||_1 \oslash ||x||_1)\}$$

Using $x^* = \mathbf{e}$, we get

$$||A \otimes x^*||_1 \oslash ||x^*||_1 = a_{n1}.$$

Also by Corollary 4.1 we get

$$\min_x(||A \otimes x||_1 \oslash ||x||_1) = a_{nn}.$$

Thus,

$$\gamma_0(j) = a_{n1} \oslash a_{nn}$$

■

Note that the 0-step approximation is not very useful as it treats the dynamic system as a static function with a single input. Therefore, every sequence has the same cost in the 0-step approximation. We provide this value as a starting point to analyze the error bound of the t-step approximation.

A. 1-step Approximation

We will now extend to the error bound of the 1-step approximation and in the next section to the t-step approximation. We will show that the error bound of the 1-step approximation is less then or equal to the error bound of the 0-step approximation. We will first define some notation.

Definition 5.1: For a job type j, we define

$$\begin{aligned} Z_i^{(1)}(j) &= [a_{i1} \oslash a_{i-1,1} \cdots a_{in} \oslash a_{i-1,n}] \otimes \mathbf{e} \\ z_i^{(1)}(j) &= a_{i1} \oslash a_{i-1,1}. \end{aligned}$$

where it is understood that each a_{ij} is from $A(j)$.

Lemma 5.1: For job type j, for any $x \in \mathbb{R}^n_{max}$, we have $x(1) = A(j) \otimes x$, and

$$z_i^{(1)}(j) \leq x(1)_i \oslash x(1)_{i-1} \leq Z_i^{(1)}(j).$$

Proof: Let $x \in \mathbb{R}^n_{max}$ be given. Then $x(1) = A \otimes x$. Suppose that $x_i(1) = a_{ij} \otimes x_j$ and $x_{i-1}(1) = a_{i-1,l} \otimes x_l$. Thus, $a_{ij} \otimes x_j \geq a_{ip} \otimes x_p$ for all $p \leq n$, and $a_{i-1,l} \otimes x_l \geq a_{i-1,p} \otimes x_p$, for all p. Through algebraic manipulation we achieve

$$\begin{aligned} x_j \oslash x_p &\geq a_{ip} \oslash a_{ij} && \forall p \leq n, \\ x_p \oslash x_l &\leq a_{i-1,l} \oslash a_{i-1,p} && \forall p \leq n. \end{aligned}$$

Using these inequalities, we prove the first inequality

$$\begin{aligned} x_i(1) \oslash x_{i-1}(1) &= (a_{ij} \otimes x_j) \oslash (a_{i-1,l} \otimes x_l) \\ &\geq (a_{ij} \otimes a_{il}) \oslash (a_{i-1,l} \otimes a_{ij}) \\ &\geq a_{i1} \oslash a_{i-1,1}. \end{aligned}$$

The second inequality follows similarly

$$\begin{aligned} x_i(1) \oslash x_{i-1}(1) &= (a_{ij} \otimes x_j) \oslash (a_{i-1,l} \otimes x_l) \\ &\leq (a_{ij} \otimes a_{i-1,l}) \oslash (a_{i-1,l} \otimes a_{i-1,j}) \\ &\leq [a_{i1} \oslash a_{i-1,1} \ldots a_{in} \oslash a_{i-1,n}] \otimes \mathbf{e} \end{aligned}$$

■

Now we will examine the error bound of the 1-step approximation. This is given by

$$\begin{aligned} \gamma_1(u_{k-1}, u_k) = \max_x\{&(||A(u_k)A(u_{k-1})x^*||_1 \\ &\oslash ||A(u_{k-1})x^*||_1) \\ &\oslash (||A(u_k)A(u_{k-1})x||_1 \\ &\oslash ||A(u_{k-1})x||_1)\} \end{aligned} \tag{6}$$

Note that $\gamma_1 \geq 0$ due to the fact the 0 is achievable by setting $x = x^*$. Because $x^* = \mathbf{e}$, the numerator of this fraction is a constant calculated from

$$\begin{aligned} ||A(u_k) \otimes A(u_{k-1}) \otimes x^*||_1 &= [a_{n1} \ldots a_{nn}](u_k) \\ &\otimes [a_{11} \ldots a_{n1}]^T(u_{k-1}) \end{aligned}$$

and

$$||A(u_{k-1}) \otimes x^*||_1 = a_{n1}(u_{k-1}).$$

Which give us

$$[a_{n1}(u_k) \otimes a_{11}(u_{k-1}) \oslash a_{n1}(u_{k-1}), \cdots, a_{nn}(u_k)] \otimes \mathbf{e}.$$

Because we ultimately want to know the difference between the 0-step approximation and the 1-step approximation, we are interested in the quantity

$$\begin{gathered} (||A(u_k) \otimes x^*||_1 \oslash ||x^*||_1) \\ \oslash (||A(u_k)A(u_{k-1})x^*||_1 \oslash ||A(u_{k-1})x^*||_1). \end{gathered}$$

Which by our previous calculations we can calculate and define

$$\begin{aligned} \delta_1 =& [a_{n1}(u_{k-1}) \oslash a_{11}(u_{k-1}), \\ & (a_{n1}(u_k) \otimes a_{n1}(u_{k-1})) \oslash (a_{n2}(u_k) \otimes a_{21}(u_{k-1})), \\ & \cdots, a_{n1}(u_k) \oslash a_{nn}(u_k)] \otimes \mathbf{e} \\ >& e. \end{aligned}$$

The action of maximizing (6) is done by minimizing the denominator. So we wish to solve

$$\min_x(||A(u_k)A(u_{k-1})x||_1 \oslash ||A(u_{k-1})x||_1).$$

This problem is similar to the zero step approximation, except that we have $A(u_{k-1})x$ everywhere we used to have just x. So now we have a similar problem, but with an extra constraint.

As when calculating

$$\min_x(||A \otimes x||_1 \oslash ||x||_1)$$

we wanted $x_n = e$ with $x_i = \epsilon$ for all other i, we want $[A(u_{k-1}) \otimes x]_n = e$ and $[A(u_{k-1}) \otimes x]_i$ as small as possible. By lemma 5.1, we know that once we fix $[A(u_{k-1}) \otimes x]_n = e$ the smallest $[A(u_{k-1}) \otimes x]_i$ can be is $e \oslash \bigotimes_{j=i+1}^n Z_j^{(1)}(u_{k-1})$. We will pick x so that

$$A(u_{k-1}) \otimes x = [e \oslash \bigotimes_{j=2}^n Z_j^{(1)}(u_{k-1}), \cdots, e]^T.$$

This value achieves

$$\begin{aligned} &\min_x(||A(u_k) \otimes A(u_{k-1}) \otimes x)||_1 \oslash ||A(u_{k-1}) \otimes x||_1) \\ =& [a_{n1}(u_k) \oslash (\bigotimes_{i=2}^n Z_i^{(1)}(u_{k-1})), \cdots, a_{nn}(u_k)] \otimes \mathbf{e}. \end{aligned}$$

Again, we are interested in the difference between this value and that of the 0-step approximation, so we will define

$$\begin{aligned}\delta_2 =& \min_x(||A(u_k) \otimes A(u_{k-1}) \otimes x||_1 \oslash ||A(u_{k-1}) \otimes x||_1) \\ &\oslash \min_x(||A(u_k) \otimes x||_1 \oslash ||x||_1) \\ =&[a_{n1}(u_k) \oslash a_{nn}(u_k) \oslash (\bigotimes_{i=2}^{n} Z_i^{(1)}(u_{k-1})), \ldots, e] \otimes \mathbf{e} \\ \geq& e.\end{aligned}$$

This proves the following theorem.

Theorem 5.2: The bound of the error of the 1-step approximation for the two job type sequence (u_1, u_2) is

$$0 \leq \gamma_1(u_1, u_2) \quad = \quad \gamma_0(u_2) \oslash \delta_1 \oslash \delta_2.$$

Thus we see that the error bound of the 1-step approximation is no worse than the error bound for the 0-step approximation.

B. t-step Approximation

We will now extend this result to the t-step approximation. We will show that the error bound for the t-step approximation is less than or equal to the error bound for the (t-1)-step approximation. The t-step approximation error bound for sequence $U = (u_1, \ldots, u_t)$ is given by

$$\begin{aligned}\gamma_t(U) \quad = \quad & \max_x(||A(u_t) \otimes \ldots \otimes A(u_1) \otimes x^*||_1 \\ & \oslash||A(u_{t-1}) \otimes \ldots \otimes A(u_1) \otimes x^*||_1) \\ & \oslash(||A(u_t) \otimes \ldots \otimes A(u_1) \otimes x||_1 \\ & \oslash||A(u_{t-1}) \otimes \ldots \otimes A(u_1) \otimes x||_1).\end{aligned} \tag{7}$$

Again it is easy to see that $\gamma_t(U) \geq 0$ due to the fact that 0 is achievable by setting $x = x^*$.

We will define a recursive value similar to that in definition 5.1 which we will use to bound $x(k)$:

Definition 5.2: Given a sequence of length t of job types, $U = (u_1, u_2, \ldots, u_t)$,

$$\begin{aligned}Z_l^{(t)}(U) \quad = \quad & \bigoplus_{\substack{d_1 \in D_Z(l-1) \\ d_2 \in D_Z(l) \\ d_1 \leq d_2}} \left(a_{ld_2}(u_t) \oslash a_{l-1,d_1}(u_t) \otimes \bigotimes_{j=d_1+1}^{d_2} Z_j^{(t-1)}(U^{t-1})\right) \\ z_l^{(t)}(U) \quad = \quad & \min_{\substack{d_1 \in D_z(l-1) \\ d_2 \in D_z(l) \\ d_1 \leq d_2}} \left(a_{ld_2}(u_t) \oslash a_{l-1,d_1}(u_t) \otimes \bigotimes_{j=d_1+1}^{d_2} z_j^{(t-1)}(U^{t-1})\right),\end{aligned}$$

with

$$\begin{aligned}D_Z(i) \quad = \quad & \left\{d \mid \forall k < d \left(a_{ik} \oslash a_{id} \leq \bigotimes_{j=k+1}^{d} Z_j^{t-1}(U^{t-1})\right)\right\} \\ D_z(i) \quad = \quad & \left\{d \mid \forall k > d \left(a_{id} \oslash a_{ik} \geq \bigotimes_{j=d+1}^{k} z_j^{t-1}(U^{t-1})\right)\right\}.\end{aligned}$$

Here we use the notation U^{t-1} to mean $(u_1, \ldots, u_{t-1})$. Moreover, we still have $Z_l^{(1)}(u_1)$ and $z_l^{(1)}(u_1)$ as in Definition 5.1.

Note that due to the monotonicity of A, Z_i is achieved with the maximum $d_1 \in D_Z(i-1)$ and $d_2 \in D_Z(i)$, and z_i is achieved with the minimum $d_1 \in D_z(i-1)$ and $d_2 \in D_z(i)$.

We will develop a lemma similar to Lemma 5.1

Lemma 5.2: Given a sequence of length t, $U = (u_1, \ldots, u_t)$, for any $x \in \mathbb{R}^n_{max}$, we have $x(t+1) = A(u_t) \otimes \ldots \otimes A(u_1) \otimes x$ with

$$z_i^{(t)}(u) \leq x_i(t+1) \oslash x_{i-1}(t+1) \leq Z_i^{(t)}(u).$$

The proof for this lemma is similar to the proof for Lemma 5.1, although slightly more complex, and is given in [13].

Using these bounds, we can prove the following result.

Lemma 5.3: Given a sequence $U = (u_1, \ldots, u_t)$,

$$A(u_t) \otimes \ldots \otimes A(u_1) \otimes e = \begin{bmatrix} x_n \oslash \left(\bigotimes_{i=2}^{n} z_i^{(t)}(U)\right) \\ \vdots \\ x_n \oslash z_n^{(t)}(U) \\ x_n \end{bmatrix}$$

The proof is supplied in [13].

We now state and prove the main theorem.

Theorem 5.3: Given a sequence of job types, $U = (u_1, \ldots, u_t)$,

$$0 \leq \gamma_t(U) \leq \gamma_{t-1}(u_2, \ldots, u_t).$$

Proof: The equation for $\gamma_t(U)$ is given in (7). To calculate $\gamma_t(U) \oslash \gamma_{t-1}(u_2, \ldots, u_t)$, which we will hereafter refer to simply as γ_t and γ_{t-1}, we will consider two parts. First we will consider

$$\begin{aligned}\phi_1 = \quad & (||A(u_t) \otimes \ldots \otimes A(u_1) \otimes x^*||_1 \\ & \oslash||A(u_{t-1}) \otimes \ldots \otimes A(u_1) \otimes x^*||_1) \\ & \oslash (||A(u_t) \otimes \ldots \otimes A(u_2) \otimes x^*||_1 \\ & \oslash||A(u_{t-1}) \otimes \ldots \otimes A(u_2) \otimes x^*||_1).\end{aligned}$$

Using Lemma 5.3 we write this as

$$\begin{aligned} & \left[a_{11}(u_t) \oslash \left(\bigotimes_{j=2}^{n} z_j^{(t-1)}(u^{t-1})\right) \cdots a_{n-1,1}(u_t) \oslash z_n^{(t-1)}(u^{t-1}), a_{nn}(u_t)\right] \otimes \mathbf{e} \\ & \oslash \left[a_{11}(u_t) \oslash \left(\bigotimes_{j=2}^{n} z_j^{(t-2)}(u_2, \ldots, u_{t-1})\right) \cdots a_{n-1,1}(u_t) \oslash z_n^{(t-2)}(u_2, \ldots, u_{t-1}), a_{nn}(u_t)\right] \otimes \mathbf{e}. \\ \leq \quad & e.\end{aligned}$$

Where the last line is a result of the fact that $z_i^{(t-1)}(U^{t-1})$ is more constrained than $z_i^{(t-2)}(u_2, \ldots, u_{t-1})$, which implies $z_i^{(t-1)}(U^{t-1}) \geq z_i^{(t-2)}(u_2, \ldots, u_{t-1})$.

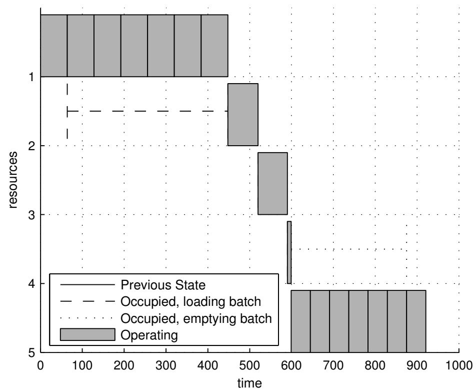

Fig. 1. A Gantt chart representing resource utilization by product 1.

The other part that we wish to consider is

$$\begin{aligned}\phi_2 = & \min_x (||A(u_t) \otimes \ldots \otimes A(u_1) \otimes x||_1 \\ & \oslash ||A(u_{t-1}) \otimes \ldots \otimes A(u_1) \otimes x||_1) \\ & \oslash \min_x (||A(u_t) \otimes \ldots \otimes A(u_2) \otimes x||_1 \\ & \oslash ||A(u_{t-1}) \otimes \ldots \otimes A(u_2) \otimes x||_1) \\ \geq & \; e.\end{aligned}$$

Where we get the final line because the first minimization is more constrained, and therefore greater than or equal to the latter minimization.

Using these two results: $\gamma_t \oslash \gamma_{t-1} = \phi_1 \oslash \phi_2 \leq 0$. ■

We have now shown that as t increases the error bound of the approximation decreases.

VI. Example

Consider a manufacturing system with 5 workstations that processes 5 job types. Consider the recipes

$$C = \begin{bmatrix} 1 & 1 & 8 & 2 & 1 \\ 7 & 12 & 8 & 4 & 70 \\ 7 & 6 & 8 & 4 & 10 \\ 7 & 1 & 4 & 1 & 2 \\ 1 & 4 & 4 & 4 & 7 \end{bmatrix}$$

$$T = \begin{bmatrix} 64 & 45 & 81 & 14 & 14 \\ 72 & 36 & 48 & 32 & 25 \\ 70 & 16 & 21 & 14 & 93 \\ 9 & 68 & 58 & 23 & 40 \\ 46 & 70 & 67 & 40 & 52 \end{bmatrix}.$$

A Gantt chart representing job type 1 is depicted in Figure 1. Note that the heap represented by job type 1 is the same form as the Gantt chart.

If we consider a quota of $\mathbf{q} = [2 \quad 2 \quad 2 \quad 2 \quad 2]^T$, we can calculate the sequence that gives the minimum makespan. There is more than one, but one sequence is given as

$$4, 4, 1, 3, 2, 2, 5, 1, 3, 5.$$

The makespan of this sequence is 8772. The maximum makespan is 9987, the mean is 9386 and the median is 9396.

By approximating the system, we obtained solutions for the 1-step and 2-step approximations. We also calculated the error bound for the 1-step approximation. The results are shown in Table I. We see that the bound on the error for the 1-step approximation is greater than the actual error.

Approximation	Optimal Sequence	Predicted Makespan	Actual Makespan	Error Bound
1-step	4,3,3,2,2, 5,1,1,5,4	8889	8836	1270
2-step	4,4,1,3,2, 2,5,1,3,5	8772	8772	784

TABLE I

The optimal sequences and makespans for the 1-step and 2-step approximations.

VII. Conclusion

We have developed a linear max-plus dynamic system representation of a batch manufacturing system model. For the max-plus representation, we showed special properties of the system. We then showed that using the approximation in [14] to solve the sequencing problem leads to a solution with bounded error from the true solution. We explicitly calculated this bound for the 0-step and 1-step approximations and showed that the error bound decreases as the complexity of the approximation increases.

References

[1] F. Blömer and H. Günther. Scheduling of a mutli-product batch process in the chemical industry. *Computers in Industry*, 36:245–259, 1998.

[2] T. Cheng, C. Ng, J. Yuan, and Z. Liu. Single machine parallel batch scheduling subject to precedence constraints. *Naval Research Logistics*, 51:949–958, 2004.

[3] X. Deng, H. Feng, G. Li, and G. Lui. A PTAS for minimizing total completion time of bounded batch scheduling. *Internation Journal of Foundations of Computer Science*, 13(6):817–827, 2002.

[4] G. Dobson and R. S. Nambimadom. The batch loading and scheduling problem. *Operations Research*, 49(1):52–65, January–February 2001.

[5] L. Dupont and C. Dhaenens-Flipo. Minimizing the makespan on a batch machine with non-identical job sizes: an exact procedure. *Computers and Operations Research*, 29(7):807–819, 2002.

[6] C. Glass, C. Potts, and V. Strusevich. Scheduling batches with sequential job processing for two-machine flow and open shops. *INFORMS Journal on Computing*, 13(2):120–137, Spring 2001.

[7] B. Heidergott, G. J. Olsder, and J. van der Woude. *Max Plus at Work: Modeling and Analysis of Synchronized Systems: A Course on Max-Plus Algebra and its Applications*. Princeton Series in Applied Mathematics. Princeton University Press, 2006.

[8] D. Hochbaum and D. Landy. Scheduling semiconductor burn-in operations to minimize total flowtime. *Operations Research*, 45(6):874–885, Nov/Dec 1997.

[9] E. Kondili, C. C. Pantelides, and R. W. H. Sargent. A general algorithm for short-term scheduling of batch operations - I. MILP formulation. *Computers and Chemical Engineering*, 17(2):211–227, 1993.

[10] B. Lin and T. Cheng. Batch scheduling in the no-wait two-machine flowshop to minimize the makespan. *Computers and Operations Research*, 28:613–624, 2001.

[11] S. H. Rich and G. J. Prokopakis. Scheduling and sequencing of batch operations in a multipurpose plant. *Ind. Eng. Chem. Process Des. Dev.*, 25:979–988, 1986.

[12] E. Sanmartí, L. Puigjaner, T. Holczinger, and F. Friedler. Combinatorial framework for effective scheduling of multipurpose batch plants. *AIChE Journal*, 48(11):2557–2570, Nov 2002.

[13] W. Weyerman and S. Warnick. An approximation method with bounded error for sequencing of batch manufacturing systems. *in preparation.*

[14] W. Weyerman, D. West, and S. Warnick. A decision-friendly approximation technique for scheduling multipurpose batch manufacturing systems. *Proc. IEEE American Control Conference*, 2006.