MAX-PLUS ALGEBRA MODELS OF QUEUEING NETWORKS

Nikolai K. Krivulin

Faculty of Mathematics and Mechanics, St.Petersburg State University, Bibliotechnaya Sq.2, Petrodvorets, St.Petersburg, 198904 Russia, E-mail: krivulin@niimm.spb.su

Keywords

Max-plus algebra, Dynamic state equation, Fork-join queueing networks, Finite buffers, Blocking of servers

Abstract

A class of queueing networks which may have an arbitrary topology, and consist of single-server fork-join nodes with both infinite and finite buffers is examined to derive a representation of the network dynamics in terms of max-plus algebra. For the networks, we present a common dynamic state equation which relates the departure epochs of customers from the network nodes in an explicit vector form determined by a state transition matrix. It is shown how the matrices inherent in particular networks may be calculated from the service times of customers. Since, in general, an explicit dynamic equation may not exist for a network, related existence conditions are established in terms of the network topology.

1 Introduction

We consider a class of queueing networks with single-server nodes and customers of a single class. In the networks, the server at each node is supplied with a buffer which may have both infinite and finite capacity. There is, in general, no restriction on the network topology; in particular, both open and closed queueing networks may be included in the class.

In addition to the ordinary service procedure, specific fork-join operations [1, 2] may be performed in each node of the networks. In fact, these operations allow customers (jobs, tasks) to be split into parts, and to be merged into one, when circulating through the network. The fork-join formalism proves to be useful in the description of dynamical processes in a variety of actual systems, including production processes in manufacturing, transmission of messages in communication networks, and parallel data processing in multi-processor systems [1]. As an example, one can consider the splitting of a message into packets, and the merging of the packets to restore the message, inherent in communication systems.

In this paper, the networks are examined so as to represent their dynamics in terms of max-plus algebra [3, 4, 5]. The max-plus algebra approach actually offers a quite compact and unified way of describing system dynamics, which may provide a useful framework for analytical study and computer simulation of discrete event systems including systems of queues.

It has been shown in [6, 7] that the evolution of both open and closed tandem queueing systems may be described by the linear algebraic equation

$$\boldsymbol{d}(k) = T(k) \otimes \boldsymbol{d}(k-1), \qquad (1)$$

where $\boldsymbol{d}(k)$ is a vector of departure epochs from the queues, $T(k)$ is a matrix calculated from service times of customers, and $\otimes$ is an operator which determines the matrix-vector multiplication in the max-plus algebra. In fact, this equation quite frequently occurs in discrete event system analysis and simulation which are based on the max-plus algebra approach. One can find a variety of related examples in [3, 8, 9, 5].

The purpose of this paper is to show that the dynamics of the networks under examination also allows of representation through dynamic state equation (1). We start with preliminary max-plus algebra definitions and related results in Section 2. Furthermore, Section 3 gives a general description of the network model, and shows how the dynamics of nodes may be described through scalar equations in terms of max-plus algebra. In Section 4, the scalar equations are extended to produce a vector representation of the dynamics of the entire network. Finally, in Section 5, explicit dynamic state equations are derived in the form of (1). Since an explicit state dynamic equation does not have to exist for an arbitrary network, related existence conditions in terms of network topology are also included in Section 5.

2 Preliminary definitions and results

We start with a brief overview of basic algebraic facts and their graph interpretation which we will exploit in the development of max-plus algebra models of queueing networks. A detailed analysis of the max-plus algebra and related algebraic systems, as well as their applications can be found in [3, 8, 4, 9, 10, 5].

Max-plus algebra is normally defined (see, e.g., [5]) as the system $\langle \underline{\mathbb{R}}, \oplus, \otimes \rangle$, where $\underline{\mathbb{R}} = \mathbb{R} \cup \{\varepsilon\}$ is the set of real numbers with $\varepsilon = -\infty$ adjoined, and the symbols $\oplus$ and $\otimes$ present binary operations determined for any $x, y \in \underline{\mathbb{R}}$ respectively as

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

As one can verify [3, 4], most of the properties of the ordinary addition and multiplication, including their associativity and commutativity, as well as distributivity of multiplication over addition, are extended to the operations $\oplus$ and $\otimes$. These properties allow usual algebraic manipulations in the max-plus algebra to be performed under the standard conventions regarding brackets and precedence of multiplication over addition. Note that, in contrast to the conventional algebra, the operation $\oplus$ is idempotent; that is, for any $x \in \underline{\mathbb{R}}$, we have $x \oplus x = x$.

There are the null and identity elements in the max-plus algebra, namely ε and $e = 0$, to satisfy the conditions $x \oplus \varepsilon = \varepsilon \oplus x = x$, and $x \otimes e = e \otimes x = x$, for any $x \in \underline{\mathbb{R}}$. The absorption rule which involves $x \otimes \varepsilon = \varepsilon \otimes x = \varepsilon$ is also true in this algebra.

The max-plus algebra of matrices is introduced in the regular way [3, 8, 4]. Specifically, for any $(n \times n)$-matrices $X = (x_{ij})$ and $Y = (y_{ij})$, the entries of $U = X \oplus Y$ and $V = X \otimes Y$ are calculated as

$$u_{ij} = x_{ij} \oplus y_{ij}, \quad \text{and} \quad v_{ij} = \sum_{k=1}^{n}{}_{\oplus}\, x_{ik} \otimes y_{kj},$$

where the symbol $\sum_{\oplus}$ denotes the iterated operation $\oplus$. As the null element, the matrix $\mathcal{E}$ with all its entries equal to ε is taken in this algebra, while the matrix $E = \mathrm{diag}(e, \ldots, e)$ with the off-diagonal entries equal to ε presents the identity element.

In perfect analogy to the conventional matrix algebra, one can define for any square matrix X,

$$X^0 = E, \quad X^q = \underbrace{X \otimes \cdots \otimes X}_{q \text{ times}} \quad \text{for } q = 1, 2, \ldots$$

Note, however, that idempotency in the max-plus algebra leads, in particular, to the identity [3]

$$(E \oplus X)^q = E \oplus X \oplus \cdots \oplus X^q.$$

Many phenomena inherent in the matrix max-plus algebra appear to be well explained in terms of their graph interpretations [3, 8, 10, 5]. To illustrate, we can consider an $(n \times n)$-matrix X with its entries $x_{ij} \in \underline{\mathbb{R}}$, and note that it can be treated as the adjacency matrix of an oriented graph with n nodes, provided each entry $x_{ij} \neq \varepsilon$ implies the existence of the arc (i, j) in the graph, whereas $x_{ij} = \varepsilon$ does the lack of the arc. The graph is then said to be *associated with* the matrix X.

Let us calculate the matrix $X^2 = X \otimes X$, and denote its entries by $x_{ij}^{(2)}$. Clearly, we have $x_{ij}^{(2)} \neq \varepsilon$ if and only if there exists at least one path from node i to node j in the graph, which consists of two arcs. Moreover, for any positive integer q, the matrix X^q has the entry $x_{ij}^{(q)} \neq \varepsilon$ only when there exists a path with the length q from i to j.

Suppose that the graph associated with the matrix X is acyclic. It is clear that we will have $X^q = \mathcal{E}$ for all $q > p$, where p is the length of the longest path in the graph. Assume now the graph not to be acyclic, and then consider any one of its circuits. Since it is possible to construct a cyclic path of any length, which lies along the circuit, we conclude that $X^q \neq \mathcal{E}$ for all $q = 1, 2, \ldots$

Finally, we consider the implicit equation in the unknown vector $\boldsymbol{x} = (x_1, \ldots, x_n)^T$,

$$\boldsymbol{x} = U \otimes \boldsymbol{x} \oplus \boldsymbol{v}, \tag{2}$$

where $U = (u_{ij})$ and $\boldsymbol{v} = (v_1, \ldots, v_n)^T$ are respectively given $(n \times n)$-matrix and n-vector. This equation actually plays a large role in max-plus algebra representations of dynamical systems including systems of queues [8, 5, 7]. The next lemma offers particular conditions for (2) to be solvable, and shows how the solution may be calculated. One can find a detailed analysis of (2) in the general case in [4].

Lemma 1 *Suppose that the entries of the matrix U and the vector $\boldsymbol{v}$ are either positive or equal to ε. Then equation (2) has the unique bounded solution $\boldsymbol{x}$ if and only if the graph associated with U is acyclic. Provided that the solution exists, it is given by*

$$\boldsymbol{x} = (E \oplus U)^p \otimes \boldsymbol{v},$$

where p is the length of the longest path in the graph.

To prove the lemma, first note that recurrent substitution of $\boldsymbol{x}$ from equation (2) into its right-hand side, made q times, and trivial manipulations give

$$\boldsymbol{x} = U^{q+1} \otimes \boldsymbol{x} \oplus (E \oplus U \oplus \cdots \oplus U^q) \otimes \boldsymbol{v}.$$

The rest of the proof may be readily furnished based on the above graph interpretation as well as on the idempotency of the operation $\oplus$.

3 The network model

In this section, we present a network model which may be considered as an extension of acyclic fork-join queueing networks investigated in [1, 2]. In fact, we do not restrict ourselves on acyclic networks, but assume the networks to have an arbitrary topology. Moreover, we examine not only the networks with the infinite capacity of buffers in their nodes, but also those with finite buffers and blocking of servers.

Proc. Intern. Workshop WODES'96, Univ. of Edinburgh, UK, Aug. 19-21, 1996, IEE: London, 1996, 76-81

3.1 A general description of the model

We consider a queueing network consisting of n single-server nodes, with customers of a single class, which circulate through the network. An example of the network under study is shown in Fig. 1.

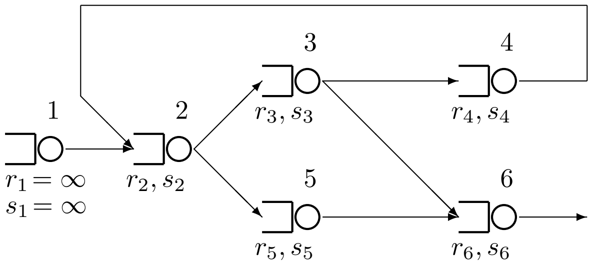

Figure 1: A queueing network with $n = 6$ nodes.

The topology of the network is described by an oriented graph $\mathcal{G} = (\mathbf{N}, \mathbf{A})$ which, in general, does not have to be acyclic. In the graph $\mathcal{G}$, the set $\mathbf{N} = \{1, \ldots, n\}$ represents the nodes of the network, and the set $\mathbf{A} = \{(i,j)\} \subset \mathbf{N} \times \mathbf{N}$ consists of arcs determining the transition routes of customers. For any $i, j \in \mathbf{N}$, the arc (i,j) belongs to $\mathbf{A}$ if and only if the ith node passes customers directly to node j.

For every node $i \in \mathbf{N}$, we introduce the set of its predecessors $\mathbf{P}(i) = \{j | (j,i) \in \mathbf{A}\}$ and the set of its successors $\mathbf{S}(i) = \{j | (i,j) \in \mathbf{A}\}$. We suppose that, in specific cases, there may be one of the conditions $\mathbf{P}(i) = \emptyset$ and $\mathbf{S}(i) = \emptyset$ encountered. Each node i with $\mathbf{P}(i) = \emptyset$ is assumed to represent an infinite external arrival stream of customers. Provided that $\mathbf{S}(i) = \emptyset$, the node is considered as an output node intended to release customers from the network.

Each node $i \in \mathbf{N}$ includes a server and its buffer which together present a single-server queue operating under the first-come, first-served (FCFS) queueing discipline. The buffer at the server in node i may have either finite or infinite capacity s_i; that is, $0 \leq s_i \leq \infty$. At the initial time, the server at each node i is assumed to be free of customers, whereas in its buffer, there may be r_i, $0 \leq r_i \leq s_i$, customers waiting for service. It is thought that the values $s_i = r_i = \infty$ are set for every node i with $\mathbf{P}(i) = \emptyset$, representing an external arrival stream of customers. We consider the numbers r_i and s_i, $i = 1, \ldots, n$, as initial conditions in the model.

To describe the dynamics of the queue in node i, we use the following symbols:

$a_i(k)$, the kth arrival epoch to the queue;

$b_i(k)$, the kth service initiation time;

$c_i(k)$, the kth service completion time;

$d_i(k)$, the kth departure epoch from the queue.

Furthermore, the service time of the kth customer at server i is denoted by τ_{ik}, $\tau_{ik} > 0$. We assume that τ_{ik} are given parameters for all $i = 1, \ldots, n$, and $k = 1, 2, \ldots$, while $a_i(k)$, $b_i(k)$, $c_i(k)$, and $d_i(k)$ are considered as unknown state variables. In addition, with the condition that the network starts operating at time zero, it is convenient to set $d_i(0) \equiv e$, and $d_i(k) \equiv \varepsilon$ for all $k < 0$, $i = 1, \ldots, n$.

As one can see, relations between the state variables, which are actually determined by the network topology, initial conditions, and special features inherent in node operation, just represent the dynamics of the network. We will describe the network dynamics in more detail and give related algebraic representations in the subsequent sections.

3.2 The dynamics of nodes

We suppose that, in addition to the usual service procedure, special join and fork operations are performed in the nodes, respectively before and after service [1]. With the condition that all buffers at servers in a network have infinite capacity, the fork-join mechanism may be described as follows.

The *join* operation is actually thought to cause each customer which comes into node i, not to enter the buffer at the server but to wait until at least one customer from every node $j \in \mathbf{P}(i)$ arrives. As soon as these customers arrive, they, taken one from each preceding node, are united to be treated as being one customer which then enters the buffer to become a new member of the queue. Note that only the customers who are waiting for service may be placed into the buffer at the node. Those customers which are ready to be joined, but, in the absence of all required customers, have not been joined yet, are assumed to reside in another place, say in an auxiliary buffer available at the node. It is suggested that the auxiliary buffers invariably have infinite capacity.

The *fork* operation at node i is initiated every time a customer releases the server after completion of his service; it consists in giving rise to several new customers to substitute for the former one. As many new customers appear in node i as there are succeeding nodes in the set $\mathbf{S}(i)$. The customers simultaneously depart the node, each going to separate node $j \in \mathbf{S}(i)$. Finally, we assume that the execution of the fork-join operations when appropriate customers are available, as well as the transition of customers within and between nodes require no time.

It is easy to see that, assuming the sets $\mathbf{P}(i)$ and $\mathbf{S}(i)$ to include no more than one node for each $i \in \mathbf{N}$, one can arrive at a queueing system in which essentially no fork-join operations are performed. As examples, both open and closed tandem systems may be considered [6, 7], which actually present queueing networks with the simplest topology.

Proc. Intern. Workshop WODES'96, Univ. of Edinburgh, UK, Aug. 19-21, 1996, IEE: London, 1996, 76-81

In order to set up the equations which represent the dynamics of nodes, let us first consider a network with infinite buffers. It follows from the above description of the fork-join mechanism that the time of the kth arrival into the queue at node i, which actually coincides with that of the completion of the kth join operation, may be represented as [1, 2]

$$a_i(k) = \begin{cases} \sum_{\oplus\, j\in\mathbf{P}(i)} d_j(k-r_i), & \text{if } \mathbf{P}(i)\neq\emptyset, \\ \varepsilon, & \text{if } \mathbf{P}(i)=\emptyset, \end{cases} \qquad (3)$$

whereas the equations which determine the other state variables are readily written in the form

$$\begin{aligned} b_i(k) &= a_i(k)\oplus d_i(k-1), & (4)\\ c_i(k) &= \tau_{ik}\otimes b_i(k), & (5)\\ d_i(k) &= c_i(k). & (6) \end{aligned}$$

Suppose now that the buffers at servers in the network may have limited capacity. In such systems, servers may be blocked according to some blocking mechanism [2, 11]. The rest of the section shows how to represent the dynamics of nodes operating under the *manufacturing* and *communication* blocking rules, both being commonly encountered in practice.

Let us first assume the network operation to follow the manufacturing blocking rule. Application of this type of blocking implies that, upon completion of his service at node i, a customer cannot release the server at the node if there is at least one succeeding node $j\in\mathbf{S}(i)$ in which the buffer is full. As soon as all nodes included in $\mathbf{S}(i)$ regain an empty buffer space, the customer leaves the server to produce new customers which have to depart node i immediately.

The inclusion of manufacturing blocking leads us to the new equation representing departure times, which is to substitute for (6),

$$d_i(k) = c_i(k)\oplus\mathcal{D}_i(k), \qquad (7)$$

where

$$\mathcal{D}_i(k) = \begin{cases} \sum_{\oplus\, j\in\mathbf{S}(i)} d_j(k-s_j-1), & \text{if } \mathbf{S}(i)\neq\emptyset, \\ \varepsilon, & \text{if } \mathbf{S}(i)=\emptyset. \end{cases}$$

Clearly, equations (3–5) remain unchanged.

Finally, we suppose that the network operates under communication blocking. This blocking rule requires the server in node i not to initiate service of a customer until there is an empty space in the buffer in each node $j\in\mathbf{S}(i)$. To represent the dynamics of node i, one may take equations (3), (5), and (6) respectively for $a_i(k)$, $c_i(k)$, and $d_i(k)$. With the symbol $\mathcal{D}_i(k)$ introduced above, an appropriate equation for $b_i(k)$ is now written as

$$b_i(k) = a_i(k)\oplus d_i(k-1)\oplus\mathcal{D}_i(k). \qquad (8)$$

4 A vector representation

We now turn to the algebraic representation of the dynamics of the entire network. To describe the dynamics in a compact form, we introduce the vectors

$$\begin{aligned} \boldsymbol{a}(k) &= (a_1(k),\ldots,a_n(k))^T,\\ \boldsymbol{b}(k) &= (b_1(k),\ldots,b_n(k))^T,\\ \boldsymbol{c}(k) &= (c_1(k),\ldots,c_n(k))^T,\\ \boldsymbol{d}(k) &= (d_1(k),\ldots,d_n(k))^T, \end{aligned}$$

and the diagonal matrix

$$\mathcal{T}_k = \begin{pmatrix} \tau_{1k} & & \varepsilon \\ & \ddots & \\ \varepsilon & & \tau_{nk} \end{pmatrix}.$$

4.1 Networks with infinite buffers

We start with the derivation of a vector representation relevant to equations (3–6) set up for networks with infinite buffers. First note that vector equations associated with (4–6) may be written immediately.

To get equation (3) in a vector form, we define $M_r = \max\{r_i|\, r_i<\infty,\, i=1,\ldots,n\}$. It is easy to see that we may now represent (3) as

$$a_i(k) = \sum_{\oplus\, m=0}^{M_r}\sum_{\oplus\, j=1}^{n} g_{ji}^m\otimes d_j(k-m),$$

where the numbers g_{ij}^m are determined using the topology of the network by the condition

$$g_{ij}^m = \begin{cases} e, & \text{if } i\in\mathbf{P}(j) \text{ and } m=r_j,\\ \varepsilon, & \text{otherwise.} \end{cases}$$

Furthermore, we introduce the matrices $G_m = \left(g_{ij}^m\right)$ for each $m=0,1,\ldots,M_r$, and then bring the above equation into its associated vector form

$$\boldsymbol{a}(k) = \sum_{\oplus\, m=0}^{M_r} G_m^T\otimes\boldsymbol{d}(k-m),$$

where G_m^T denotes the transpose of the matrix G_m. Note that each matrix G_m presents an adjacency matrix of the partial graph $\mathcal{G}_m = (\mathbf{N},\mathbf{A}_m)$ with $\mathbf{A}_m = \{(i,j)|\, i\in\mathbf{P}(j),\, r_j=m\}$.

We are now in a position to describe the network dynamics in vector terms. By replacing equations (3–6) with their vector representations, we obtain

$$\begin{aligned} \boldsymbol{a}(k) &= \sum_{\oplus\, m=0}^{M_r} G_m^T\otimes\boldsymbol{d}(k-m),\\ \boldsymbol{b}(k) &= \boldsymbol{a}(k)\oplus\boldsymbol{d}(k-1),\\ \boldsymbol{c}(k) &= \mathcal{T}_k\otimes\boldsymbol{b}(k),\\ \boldsymbol{d}(k) &= \boldsymbol{c}(k). \end{aligned}$$

Clearly, these equations can be reduced to an equation in one vector variable, say $\boldsymbol{d}(k)$. In that case, appropriate substitutions will lead us to the equation

$$\begin{aligned}\boldsymbol{d}(k) &= \mathcal{T}_k \otimes G_0^T \otimes \boldsymbol{d}(k) \oplus \mathcal{T}_k \otimes \boldsymbol{d}(k-1) \\ &\quad \oplus \mathcal{T}_k \otimes \sum_{m=1}^{M_r}{}_{\oplus} G_m^T \otimes \boldsymbol{d}(k-m). \qquad (9)\end{aligned}$$

4.2 Networks with finite buffers

Consider a network with finite buffers, and assume that it operates under the manufacturing blocking rule. With $M_s = \max\{s_i + 1 | s_i < \infty, i = 1, \dots, n\}$, equation (7) may be put in the form

$$d_i(k) = c_i(k) \oplus \sum_{m=1}^{M_s}{}_{\oplus} \sum_{j=1}^{n}{}_{\oplus} h_{ij}^m \otimes d_j(k-m),$$

where

$$h_{ij}^m = \begin{cases} e, & \text{if } j \in \mathbf{S}(i) \text{ and } m = s_j + 1, \\ \varepsilon, & \text{otherwise.} \end{cases} \qquad (10)$$

In a similar way as in Section 4.1, one can introduce the matrices $H_m = \left(h_{ij}^m\right)$, $m = 1, \dots, M_s$, and then rewrite (7) so as to get a representation for $\boldsymbol{d}(k)$.

Taking into account that the equations set up previously to represent the vectors $\boldsymbol{a}(k)$, $\boldsymbol{b}(k)$, and $\boldsymbol{c}(k)$ remain valid, we arrive at the set of equations

$$\begin{aligned}\boldsymbol{a}(k) &= \sum_{m=0}^{M_r}{}_{\oplus} G_m^T \otimes \boldsymbol{d}(k-m), \\ \boldsymbol{b}(k) &= \boldsymbol{a}(k) \oplus \boldsymbol{d}(k-1), \\ \boldsymbol{c}(k) &= \mathcal{T}_k \otimes \boldsymbol{b}(k), \\ \boldsymbol{d}(k) &= \boldsymbol{c}(k) \oplus \sum_{m=1}^{M_s}{}_{\oplus} H_m \otimes \boldsymbol{d}(k-m).\end{aligned}$$

Without loss of generality, we consider that $M_r = M_s = M$. If it actually holds that $M_r < M_s$ (the inequality $M_r > M_s$ is contradictory to the initial conditions), one may set $M = M_s$, and then define $G_m = \mathcal{E}$ for all $m = M_r + 1, M_r + 2, \dots, M_s$. With this assumption, we may drop the subscripts so as to write M instead of both M_r and M_s.

Proceeding to an equation in $\boldsymbol{d}(k)$, we get

$$\begin{aligned}\boldsymbol{d}(k) &= \mathcal{T}_k \otimes G_0^T \otimes \boldsymbol{d}(k) \oplus \mathcal{T}_k \otimes \boldsymbol{d}(k-1) \\ &\quad \oplus \sum_{m=1}^{M}{}_{\oplus} (\mathcal{T}_k \otimes G_m^T \oplus H_m) \otimes \boldsymbol{d}(k-m). \qquad (11)\end{aligned}$$

Let us now assume the network to follow the communication blocking rule. In the same way as for manufacturing blocking, one may define matrices $H_1, \dots, H_M$ through (10), and represent equation (8) in its vector form. The set of vector equations describing the network dynamics then becomes

$$\begin{aligned}\boldsymbol{a}(k) &= \sum_{m=0}^{M}{}_{\oplus} G_m^T \otimes \boldsymbol{d}(k-m), \\ \boldsymbol{b}(k) &= \boldsymbol{a}(k) \oplus \boldsymbol{d}(k-1) \oplus \sum_{m=1}^{M}{}_{\oplus} H_m \otimes \boldsymbol{d}(k-m), \\ \boldsymbol{c}(k) &= \mathcal{T}_k \otimes \boldsymbol{b}(k), \\ \boldsymbol{d}(k) &= \boldsymbol{c}(k).\end{aligned}$$

Finally, by combining these equations, we have

$$\begin{aligned}\boldsymbol{d}(k) &= \mathcal{T}_k \otimes G_0^T \otimes \boldsymbol{d}(k) \oplus \mathcal{T}_k \otimes \boldsymbol{d}(k-1) \\ &\quad \oplus \mathcal{T}_k \otimes \sum_{m=1}^{M}{}_{\oplus} (G_m^T \oplus H_m) \otimes \boldsymbol{d}(k-m). \qquad (12)\end{aligned}$$

5 The explicit state equation

Let us consider equations (9), (11), and (12) derived above, and note that they actually present implicit equations in the system state variable $\boldsymbol{d}(k)$. In this section, we show how these equations may be put in their associated explicit forms which are normally more suitable for analytical treatments and computer simulation of the network dynamics. Since, in general, the implicit equations do not have to be explicitly solvable, the conditions for an explicit state equation to exist are also established.

In order to examine the implicit equations, we first note that they all take the form of (2) with $U = \mathcal{T}_k \otimes G_0^T$. Since the matrix $\mathcal{T}_k$ is diagonal, each graph associated with the matrix G_0^T will be likewise associated with U. In addition, the graph associated with the matrix G_0 and that with its transpose are both acyclic or not at once. Finally, both graphs have a common length p of their longest paths.

Now it is not difficult to apply Lemma 1 so as to prove the following statement.

Theorem 1 *Suppose that in the network model with infinite buffers, the graph $\mathcal{G}_0$ associated with the matrix G_0 is acyclic. Then equation* (9) *can be solved to produce the explicit state dynamic equation*

$$\boldsymbol{d}(k) = \sum_{m=1}^{M}{}_{\oplus} T_m(k) \otimes \boldsymbol{d}(k-m), \qquad (13)$$

with the state transition matrices

$$\begin{aligned}T_1(k) &= (E \oplus \mathcal{T}_k \otimes G_0^T)^p \otimes \mathcal{T}_k \otimes (E \oplus G_1^T), \\ T_m(k) &= (E \oplus \mathcal{T}_k \otimes G_0^T)^p \otimes \mathcal{T}_k \otimes G_m^T, \\ &\quad m = 2, \dots, M,\end{aligned}$$

where p is the length of the longest path in $\mathcal{G}_0$.

As one can see, the matrix coefficient at $\boldsymbol{d}(k)$ on the right-hand side of both equation (11) and (12), which is just responsible for explicit representation (13) to exist, remains the same as in equation (9) set up for the network with infinite buffers. We may therefore conclude that entering finite buffers into the network model has no effect on the existence of its associated explicit dynamic state equation.

Now one can reformulate Theorem 1 to extend it to the networks with finite buffers. In short, under the same conditions as in the theorem, equations (11) and (12) may be put in the form of (13), with the state transition matrices defined respectively as

$$\begin{aligned} T_1(k) &= (E \oplus \mathcal{T}_k \otimes G_0^T)^p \otimes (\mathcal{T}_k \oplus \mathcal{T}_k \otimes G_1^T \oplus H_1), \\ T_m(k) &= (E \oplus \mathcal{T}_k \otimes G_0^T)^p \otimes (\mathcal{T}_k \otimes G_m^T \oplus H_m), \\ & \quad m = 2, \ldots, M, \end{aligned}$$

and

$$\begin{aligned} T_1(k) &= (E \oplus \mathcal{T}_k \otimes G_0^T)^p \otimes \mathcal{T}_k \otimes (E \oplus G_1^T \oplus H_1), \\ T_m(k) &= (E \oplus \mathcal{T}_k \otimes G_0^T)^p \otimes \mathcal{T}_k \otimes (G_m^T \oplus H_m), \\ & \quad m = 2, \ldots, M. \end{aligned}$$

Finally, with the extended state vector

$$\widehat{\boldsymbol{d}}(k) = \begin{pmatrix} \boldsymbol{d}(k) \\ \boldsymbol{d}(k-1) \\ \vdots \\ \boldsymbol{d}(k-M+1) \end{pmatrix},$$

we may bring (13) into the form of (1):

$$\widehat{\boldsymbol{d}}(k) = \widehat{T}(k) \otimes \widehat{\boldsymbol{d}}(k-1),$$

where the new state transition matrix is defined as

$$\widehat{T}(k) = \begin{pmatrix} T_1(k) & T_2(k) & \cdots & \cdots & T_M(k) \\ E & \mathcal{E} & \cdots & \cdots & \mathcal{E} \\ & \ddots & \ddots & & \vdots \\ & & \ddots & \ddots & \vdots \\ \mathcal{E} & & & E & \mathcal{E} \end{pmatrix}.$$

In conclusion, let us assume that for a network, the partial graph $\mathcal{G}_0$ has a circuit. We may consider the network depicted in Fig. 1 as an appropriate illustration if we put $r_2 = r_3 = r_4 = 0$. In that case, there is the circuit in the graph $\mathcal{G}_0$, including nodes 2, 3, and 4. Then, as it follows from Lemma 1, the implicit dynamic equation associated with the network cannot be solved in an explicit form.

One can see, however, that it is easy to make the equation solvable only by setting new initial conditions, without changing the network topology. Since $\mathcal{G}_0 = (\mathbf{N}, \mathbf{A}_0)$, where $\mathbf{A}_0 = \{(i,j) | \, i \in \mathbf{P}(j), \, r_j = 0\}$, we may eliminate arcs from the graph $\mathcal{G}_0$ by substituting nonzero values for some parameters r_i set initially to 0. One can compare the network in Fig. 1, under the conditions $r_2 = r_3 = r_4 = 0$, with that subject to $r_2 = r_4 = 0$, and $r_3 = 1$, as an example.

References

[1] François Baccelli and Armand M. Makowski, "Queueing models for systems with synchronization constraints", *Proceedings of the IEEE*, vol. 77, no. 1, pp. 138–160, 1989.

[2] Albert G. Greenberg, Boris D. Lubachevsky, and Isi Mitrani, "Algorithms for unboundedly parallel simulations", *ACM Transactions on Computer Systems*, vol. 9, no. 3, pp. 201–221, 1991.

[3] Raymond Cuninghame-Green, *Minimax Algebra*, vol. 166 of *Lecture Notes in Economics and Mathematical Systems*, Springer-Verlag, Berlin, 1979.

[4] Guy Cohen, Pierre Moller, Jean-Pierre Quadrat, and Michel Viot, "Algebraic tools for the performance evaluation of discrete event systems", *Proceedings of the IEEE*, vol. 77, no. 1, pp. 39–58, 1989.

[5] Christos G. Cassandras, Stéphane Lafortune, and Geert Jan Olsder, "Introduction to the modelling, control and optimization of discrete event systems", Tech. Rep. 95–92, Faculty of Technical Mathematics and Informatics, Delft University of Technology, 1995.

[6] Nikolai K. Krivulin, "Using max-algebra linear models in the representation of queueing systems", in *(Proceedings of the 5th SIAM Conference on Applied Linear Algebra, June 15–18, 1994, Snowbird, UT)*, John G. Lewis, Ed., Philadelphia, PA, 1994, SIAM, pp. 155–160.

[7] Nikolai K. Krivulin, "A max-algebra approach to modeling and simulation of tandem queueing systems", *Mathematical and Computer Modelling*, vol. 22, no. 3, pp. 25–37, 1995.

[8] Guy Cohen, Didier Dubois, Jean-Pierre Quadrat, and Michel Viot, "A linear-system-theoretic view of discrete-event processes and its use for performance evaluation in manufacturing", *IEEE Transactions on Automatic Control*, vol. AC-30, no. 3, pp. 210–220, 1985.

[9] François Baccelli and Miguel Canales, "Parallel simulation of stochastic petri nets using recurrence equations", *ACM Transactions on Modeling and Computer Simulation*, vol. 3, no. 1, pp. 20–41, 1993.

[10] Victor P. Maslov and Vasiliy N. Kolokoltsov, *Idempotent Analysis and Its Applications to Optimal Control Theory*, Nauka, Moscow, 1994, (in Russian).

[11] Dinah W. Cheng, "On the design of a tandem queue with blocking: Modeling, analysis, and gradient estimation", *Naval Research Logistics*, vol. 41, pp. 759–770, 1994.