BATCH DETERMINISTIC AND STOCHASTIC PETRI NETS: MODELLING, ANALYSIS AND APPLICATION TO INVENTORY SYSTEMS

K. Labadi, H. Chen, L. Amodeo and C. Chu

ISTIT- Industrial Systems Optimization Group, CNRS (FRE 2732)
UTT- 12 rue Marie Curie, BP 2060, 10010 Cedex, France

Abstract: We recently introduced a new stochastic Petri net model called "batch deterministic and stochastic Petri nets" (BDSPNs) capable of describing the synchronization of discrete and batch token flows in discrete batch processes. It is a powerful formal model for the study of inventory systems and supply chains where materials are processed or ordered in finite discrete quantities (batches) and many operations such as inventory replenishment, manufacturing and distribution are usually performed in a batch way because of the batch nature of customer orders and/or in order to take advantages of the economies of scale. In this paper, the modelling and analysis powers of the model are demonstrated through inventory systems. We show particularly how to model and evaluate the performance of a continuous review (s, S) inventory system with stochastic and batch demand using BDSPNs.

Keywords: Petri nets, modelling, performance evaluation, inventory systems, supply chains.

1. INTRODUCTION

Discrete batch processes play an important role in industry. It exists in manufacturing systems, inventory systems and supply chains where materials are processed in finite quantities (batches), and many operations such as inventory replenishment, manufacturing and distribution are usually performed in a batch way because of the batch nature of customer orders and/or in order to take advantages of the economies of scale. Discrete properties of batch processes render their design and operation extraordinarily difficult, and pose challenging issues in system design and operation control. Issues such as analysis and optimization of these processes need a relevant batch model and analysis methods which can grasp the batch nature of the systems. The model should be able to explicitly describe the batch sizes of different operations in a batch process.

Petri nets have been recognized as a powerful tool for modelling and analysis of discrete event systems. While Petri nets have been studied for more than 40 years (Zurawski et al. 1994), they played a relatively minor role in modelling and analysis of discrete batch processes in inventory systems and supply chains. Supply chains are modelled by using colored Petri nets (Van der Aalst et al. 1992), where every entity in the chain is modelled by a block with action, resource, and control, which is a subnet of the colored Petri net model. Supply chains are also modelled by using generalized stochastic Petri nets (Viswanadham et al. 2000). For inventory systems with independent demand, entities of supply chain, they are modelled by using First-order hybrid Petri nets that combine fluid and discrete event dynamics (Balduzzi et al. 2000; Furcas et al. 2001). These Petri net models, however, ignore an important feature of these systems. That is, operations such as inventory replenishment and distribution are usually performed in a batch way. We recently introduced a new stochastic Petri net model called "Batch Deterministic and Stochastic Petri nets" (BDSPNs) capable of describing the synchronization of discrete and batch token flows in discrete batch processes. It is a powerful formal tool for modelling and analysis of inventory systems and supply chains (Chen et al. 2002, 2003). Our model is a discrete Petri net model, rather than a hybrid Petri net model. It thus keeps the simplicity of discrete Petri nets.

In the remainder of this paper, the model is introduced, its novelty for modelling discrete batch processes is illustrated, and some fundamental concepts and analysis methods of the model are presented. As application, a continuous review (*s*, *S*) inventory system with stochastic and batch demand and stochastic transportation time, is modelled and its performance is evaluated analytically using BDSPNs.

2. THE BDSPN MODEL

BDSPNs are developed by extending Deterministic and stochastic Petri nets (DSPNs) (Lindemann, 1998) with batch places and batch tokens. In a BDSPN, there are two types of places, discrete places and batch places. Tokens in a discrete place are viewed indifferently as in standard Petri nets, while tokens in a batch place, called batch tokens, may have different sizes and are viewed as different individuals. Different ways are used to represent the marking (state) of a discrete place and the marking of a batch place. The first marking is represented by a nonnegative integer as in standard Petri nets, while the second marking is represented by a set of nonnegative integers. The set may contain identical elements (multi-set) and each integer in the set represents a batch token with a given size. We use a vector μ to represent the marking of a BDSPN, where μ(p) is a nonnegative integer for a discrete place p and a set of nonnegative integers for a batch place p. This marking, called μ-marking, represents the state of the BDSPN. Moreover, for defining BDSPNs, another type of marking, called M-marking, is also introduced. For each discrete place, its M-marking is the same as its μ-marking, while for each batch place its M-marking is defined as the total size of the batch tokens in the place.

2.1 Formal definition.

Formally, a batch deterministic and stochastic Petri net (BDSPN) is specified as a nine-tuple:

$$\text{BDSPN}= (P, T, I, O, V, W, \Pi, D, \mu_0) \quad (1)$$

where:

$P = P_d \cup P_b$ is a finite set of places consisting of discrete places in P_d and batch places in P_b.

$T = T_i \cup T_d \cup T_e$ is a finite set of transitions consisting of immediate transitions in T_i, deterministic transitions in T_d, and exponentially distributed transitions in T_e.

$I \subseteq (P\times T)$, $O \subseteq (T\times P)$, and $V \subseteq (P\times T_i)$ define the input arcs, the output arcs and the inhibitor arcs of the transitions, respectively. It is assumed that only immediate transitions are associated with inhibitor arcs, i.e., $V \subseteq (P\times T_i)$, and the inhibitor arcs and the input arcs are two disjoint sets.

W defines the weights of the input, output and inhibitor arcs that may depend on the current M-marking of the net.

Π is the firing priority function which assigns a priority to each transition. It is assumed that timed transitions have the lowest priority, i.e., $\Pi(t) = 0$ if $t \in T_d \cup T_e$. For immediate transitions, $\Pi(t) \geq 1$.

D defines the firing delay associated with each timed transition. Each deterministic timed transition $t \in T_d$ is associated with a constant firing delay and each exponentially distributed timed transition $t \in T_e$ is associated with a mean firing rate λ_i.

μ_0 is the initial μ-marking of the net.

In graphical representation of the net, discrete places and batch places are represented by single circles and squares with an embedded circle, respectively. Immediate, deterministic, and exponentially distributed transitions are represented by thin bars, filled rectangles, and empty rectangles, respectively. Inhibitor arcs are represented by arrows ending with a small circle. A transition t is said to be a batch transition (resp. a discrete transition) if it has at least one input batch place (resp. if it has no input batch place). Discrete tokens are represented by dots, while batch tokens are represented by Arabic numbers that indicate their sizes (see Fig.1). In the following, the set of input places, the set of output places, and the set of inhibitor places of transition t are denoted by $^{\bullet}t$, $t^{\bullet}$, and $^{\circ}t$, respectively, where $^{\bullet}t = \{ p \mid (p, t) \in I \}$, $t^{\bullet} = \{p \mid (t, p) \in O\}$, and $^{\circ}t = \{p \mid (p, t) \in V\}$. We denote by $w(i, j)$ the weight of arc (i, j) in the net.

2.2 Enabling and Firing rules.

Two types of transition firing called "*batch firing*" and "*discrete firing*" govern the state evolution of the net and synchronize discrete and batch token flows.

Discrete Enabling and Firing. In this case, a transition *t* has no batch input place. It is said to be enabled at μ-marking μ (its corresponding M-marking *M*) if and only if:

$$\forall p \in {}^{\bullet}t, \qquad M(p) \geq w(p,t) \quad (2)$$

$$\forall p \in {}^{\circ}t, \qquad M(p) < w(p,t) \quad (3)$$

The discrete firing of *t* leads to a new μ-marking μ':

$$\forall p \in {}^{\bullet}t: \qquad \mu'(p) = \mu(p) - w(p,t) \quad (4)$$

$$\forall p \in t^{\bullet} \cap P_d: \qquad \mu'(p) = \mu(p) + w(t,p) \quad (5)$$

$$\forall p \in t^{\bullet} \cap P_b: \quad \mu'(p) = \mu(p) + \{w(t,p)\} \quad (6)$$

The firing rules (4) and (5) are the same as those for a transition in a standard Petri net. For each output batch place *p*, after the firing of transition *t*, a batch token with the size equal to the weight *w*(*p, t*) will be created ((6)).

Batch Enabling and Firing. In this case, a transition *t* has at least one input batch place. It is said to be enabled at μ-marking μ if and only if there is a *batch firing index* (positive integer) $q \in$ IN ($q>0$) such that:

$$\forall p \in {}^{\bullet}t \cap P_b, \exists b \in \mu(p): \quad q = b / w(p,t) \quad (7)$$

$$\forall p \in {}^{\bullet}t \cap P_d, \qquad M(p) \geq q \times w(p,t) \qquad (8)$$

$$\forall p \in {}^{\circ}t, \qquad M(p) < w(p,t) \qquad (9)$$

The batch firing of *t* leads to a new μ-marking μ':

$$\forall p \in {}^{\bullet}t \cap P_d : \mu'(p) = \mu(p) - q \times w(p,t) \qquad (10)$$

$$\forall p \in {}^{\bullet}t \cap P_b : \mu'(p) = \mu(p) - \{q \times w(p,t)\} \qquad (11)$$

$$\forall p \in t^{\bullet} \cap P_d : \mu'(p) = \mu(p) + q \times w(t,p) \qquad (12)$$

$$\forall p \in t^{\bullet} \cap P_b : \mu'(p) = \mu(p) + \{q \times w(t,p)\} \qquad (13)$$

A batch transition is enabled if and only if: 1) Each batch input place of the transition has a batch token with the batch firing index *q* common to all batch input places ((7)), 2) Each discrete input place of the transition has enough tokens to simultaneously fire the transition for a number of times given by the index ((8)), and 3) The number of tokens in each inhibitor place of the transition is less than the weight of the corresponding inhibitor arc ((9)). For any batch output place (resp. discrete output place), the firing of the transition generates a batch token with the size (resp. a number of discrete tokens with the number) given by the product of the batch firing index and the weight of the corresponding arc.

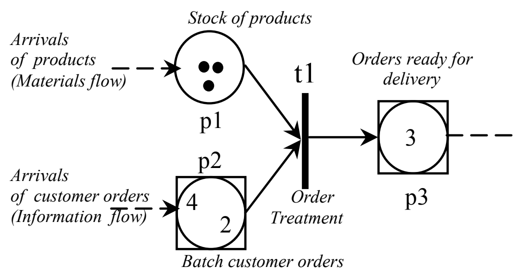

Fig. 1. Order processing example.

To well understand the meaning of batch firing concept of BDSPN model, consider the net in Fig. 1. Customer orders with *different sizes* arrive and are recorded in batch place *p2* where they wait for treatment by batch transition *t1*. To fill a given customer order with size *b*, we need a number of products from the stock represented by *p1*. In this example, a batch token with size *b* needs *b* discrete tokens since *w(p1, t1) = 1*. For instance, to meet a customer order of size 2 (a batch token with *b = 2* in *p2*), we need *q*×*w*(*p1, t1*) of products, where *q* is the batch firing index of *t1* given by *q = b/w(p2, t1) = b = 2*. At the current μ-marking $\mu = (3, \{4, 2\}, 3)^T$, *t1* is enabled with *q = 2*. The batch firing of *t1* with the index will remove the batch token from *p2* and *q×w(p1, t) = 2* discrete tokens from *p1*. A batch token with size *q×w(t, p4) = 2* will be created in *p3*.

2.3 Temporal behaviour.

When both timed and immediate transitions are enabled at a μ-marking, it is assumed that only immediate transitions can be fired because timed transitions have the lowest priority. When immediate transitions with different priorities are enabled, only those with the highest priority can be fired. When several conflicting transitions with the same highest priority are enabled, each of them has the same probability to be fired. Each immediate transition is fired in zero time whereas each timed transition is fired after either a deterministic or an exponentially distributed firing delay. When some timed transitions are enabled at a μ-marking, the transition with the minimum firing delay will cause a change of the μ-marking. Further, as in DSPNs, it is assumed that after the change of the μ-marking each timed transition newly enabled samples a remaining firing time from its firing delay distribution. Each timed transition which was enabled in the previous μ-marking and is still enabled in the current marking keeps its remaining firing time (Ajmone Marsan et al. 1995).

In addition, particular policies should be specified to choose a batch token in each batch input place to fire its output transition. In a BDSPN, batch tokens are distinguishable by their sizes and are viewed as different individuals. In each batch place, various batch tokens may exist and be taken as candidates for the firing of its output transitions. If multiple batch tokens with different sizes can fire a transition, one batch token has to be chosen for the firing of the transition. Possible selection policies include FIFO rule which respects the arrival order of batch tokens in each batch place, random rule, and token size-based rule.

3. MODELLING OF INVENTORY SYSTEMS

In this section, we discuss how BDSPNs can be used to model inventory systems which are an important component of supply chains. Inventory systems with independent demand may use fixed or variable order quantity policies based on either continuous or periodic review of inventory position. The inventory position is defined as on-hand inventory plus outstanding orders minus backorders. Fixed order quantity policies place an order of fixed size whenever the inventory position of a stock falls to a pre-specified level, while variable order quantity policies place an order of variable size at regular intervals to raise the inventory position to a pre-specified value. The most frequently used inventory policies include order-up-to-level policy, batch ordering policy (*R, Q*), and (*s, S*) policy. As an example, we consider an inventory system with continuous review (*s*, *S*) policy.

For this system, when the inventory position *IP* drops below a given reorder point *s*, an order with quantity *S – IP* will be placed to raise the inventory position to a given order-to-level *S*. Figure 2 shows the BDSPN models of the inventory system: model (a) with continuous review and backorders, and model (b) with periodic review and no backorder. In model (a), place p_1 represents on-hand (physical) inventory of the stock considered. Batch place p_3 represents outstanding orders (the orders placed by the stock but their corresponding shipments not received yet). Discrete place p_4 represents backorders (unfilled customer demands). Place p_2 represents on-hand inventory of the stock plus its outstanding orders, that is, $M(p_2) = M(p_1) + M(p_3)$. The inventory position IP equals $M(p_1) + M(p_3) - M(p_4) = M(p_2) - M(p_4)$. Inventory replenishment decisions are made based on the position. In the model, customer demand is assumed to be a Poisson process, which is specified by transition t_4 whose firing time is subject to an exponential distribution. Customer demand will be filled if there is sufficient on-hand inventory. Otherwise, the demand will be backordered. The fulfilment of customer demand will decrease on-hand inventory as well as the size of backorders. This is described by the arcs from places p_1, p_4 and p_2 to transition t_1. When the inventory position $M(p_2) - M(p_4)$ drops below the reorder point s, i.e., $M(p_2) - M(p_4) < s$ or equivalently $M(p_2) < s + M(p_4)$, an order with quantity $S - IP = S - M(p_2) + M(p_4)$ will be placed to the supplier. The placement of the order will increase the size of outstanding orders by this quantity. This is described by immediate transition t_3, its associated arcs to places p_2 and p_3, and the weights of the arcs. When the M-marking of place p_2, $M(p_2)$ is less than $s + M(p_4)$, transition t_3 will be fired, which generates a batch token with size $S - M(p_2) + M(p_4)$ in place p_3 and $S - M(p_2) + M(p_4)$ discrete tokens in place p_2. If there is a batch token in place p_3, a replenishment event represented by the firing of transition t_2 will occur, which delivers the corresponding order to the stock. The firing delay of transition t2 represents the transportation delay of the order.

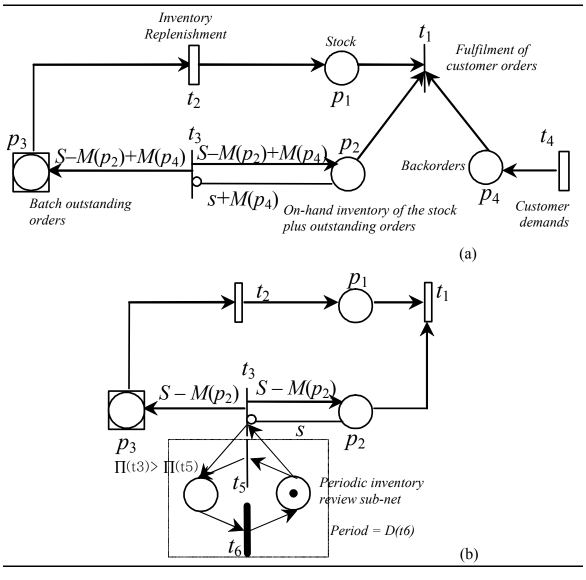

Fig. 2. BDSPN models of (s, S) inventory systems.

For modelling the system without backorder, the place p_4 used for representing backorders in (a) is removed, exponential transition t_4 and immediate transition t_1 are merged to a new exponential transition t_1, the term $M(p_4)$ in the weights of the arcs associated with transition t_3 is removed accordingly. For modelling the periodic review feature of the system, in model (b), an additional subnet used for describing the feature is introduced and coupled with the net in (a) through a deterministic transition t_3 with $\Pi(t3) > \Pi(t5)$. Our model can also describe inventory systems with order-up-to level policy and batch ordering policy.

4. PERFORMANCE EVALUATION APPROACH

One ultimate goal for the introduction of BDSPN model is to evaluate the performance of discrete event systems with batch behaviours. Similar to existing stochastic Petri nets, the main performance analysis technique for BDSPNs is based on the analysis of the stochastic marking process of the net $\{\mu(t),\ t \geq 0\}$. In the case of a bounded net, the resulting process may be a continuous-time Markov chain, a semi-Markov chain, a Markov regenerative process, or a generalized semi-Markov process depending on whether there are deterministic transitions and on whether a deterministic transition can be fired concurrently with other timed transitions. For the first three Markov chains or process, there are efficient analysis methods. Having known the marking process, some important performance indexes of the net such as average number of tokens in a place and average firing frequency of a transition can then obtained from the steady state distribution of the underlying Markov chain or process. The performance evaluation can be done following the steps: 1) The μ-reachability graph is generated from the BDSPN model, 2) The μ-reachability graph is converted to its corresponding stochastic process, 3) The stochastic process is identified, analyzed and solved analytically or numerically to obtain its steady state probabilities, 4) The performance indexes of the system are computed using the solution (steady state probabilities) of the stochastic process. In the case of an unbounded net, simulation or approximation methods are required.

BDSPN Performance Indexes. The performance of a BDSPN can be expressed in terms of performance indexes. These indexes can be computed using a unifying approach if they are defined over the reachable μ-marking set $R(N, \mu_0)$ of the model and the mean value of each index can then be derived using the steady-state probabilities π of the model as explained in the following.

The *mean number of batch tokens* in a give batch place is given by:

$$\bar{\mu}(p) = \sum_{i:\mu_i \in R(N,\mu_0)} card\left(\mu_i(p)\right) \cdot \pi_i \tag{14}$$

where $card(\mu_i(p))$ is the cardinality of set $\mu_i(p)$.

The *mean number of batch tokens with the size equal to b* in a given batch place is given by:

$$\bar{\mu}(p) = \sum_{i:\mu_i \in R(N,\mu_0) \wedge \mu_i(p)=b} card\left(\mu_i(p)\right) \cdot \pi_i \tag{15}$$

The *mean number of tokens* in a given discrete place is given by:

$$\bar{\mu}(p) = \sum_{i:\mu_i \in R(N,\mu_0)} \mu_i(p) \cdot \pi_i \tag{16}$$

The *mean total size of tokens* in a given batch place is given by:

$$\bar{M}(p) = \sum_{i:\mu_i \in R(N,\mu_0)} \left(\sum_{b \in \mu_i(p)} b \right) \cdot \pi_i \tag{17}$$

Let $S(t_{j[q]})$ be the set of μ-markings at which batch transition t_j is fired with batch firing index q, the firing frequency of the transition with this index can be computed by:

$$F(t_{j[q]}) = \sum_{i:\mu_i \in S(t_{j[q]})} \lambda_{j[q]} \pi_i \qquad (18)$$

where $\lambda_{j[q]}$ is the firing rate of t_j with index q.

5. PERFORMANCE EVALUATION OF (*s*, *S*) INVENTORY SYSTEM

In the section, we model and analyze a continuous review (*s*, *S*) inventory system with Poisson and batch demand and exponential transportation time. The system is modelled using a BDSPN in Fig. 3. Compared to the model presented in Fig. 2(b), the batch place p4 connected to exponential transition *t1*, is used to model the source of batch customer orders. In the BDSPN, we assume that transition t1 generates two different sizes of batch customer orders (with the batch sizes 1 and 2 respectively) which are specified by the μ-marking of p4 (i.e.; μ(p4) = {1, 2}). It is assumed that the reorder point and the order-up-to-level of the inventory system are taken as $s = 4$ and $S = 6$ respectively, and the initial μ-marking of the net is $\mu_0 = (6, 6, \varnothing, \{1, 2\})$.

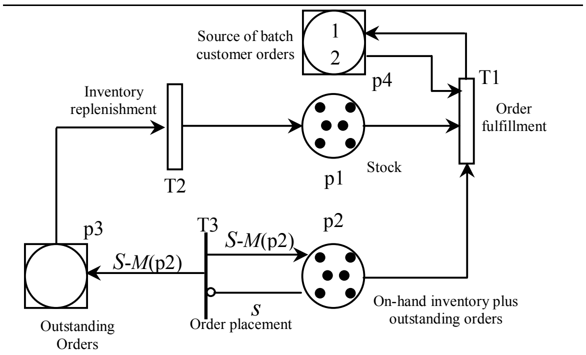

Fig. 3. BDSPN model of an (*s*, *S*) inventory system with stochastic and batch demand

The state space of the net is represented by its μ-reachability graph shown in Fig. 4. In the graph, each directed edge is associated with a label representing the transition whose firing generates the successor μ-marking. Each batch transition is marked by its corresponding batch firing index *q*. The μ-markings obtained can be classified into *vanishing* and *tangible* μ-markings. A vanishing μ-markings is one in which at least one immediate transition is enabled, and a tangible μ-marking is one in which no immediate transition is enabled. In the μ-reachability graph, the vanishing μ-markings μ_i (i = *0 to 9*) are represented by rectangles and two tangible μ-markings μ and μ' are represented by dotted rectangles. After eliminating the vanishing μ-markings by merging them with their successor tangible μ-markings and converting the reduced μ-reachability graph to its corresponding stochastic process, we get a Continuous Timed Markov Chain (CTMC) represented in Fig. 5. Assuming that the firing delays of batch transitions *t1* and *t2* (the demand rate and the inventory replenishment rate) are exponentially distributed with rates $\lambda 1_{[q]} = \lambda 1$ and $\lambda 2_{[q]} = \lambda 2$ respectively for any feasible batch firing index *q*, the infinitesimal generator matrix (transition rate matrix) denoted by *A* is given in Fig. 6.

By solving the linear system $\pi.A = 0$, and $\Sigma_i \pi_i = 1$, the steady-state probabilities π can be explicitly obtained as functions of parameters λ_1 and λ_2 given in Table 1.

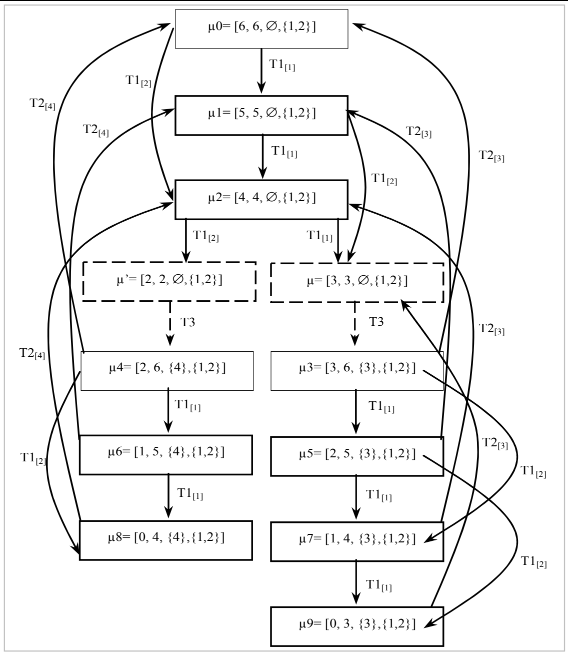

Fig.4. The μ-marking reachability graph of the BDSPN.

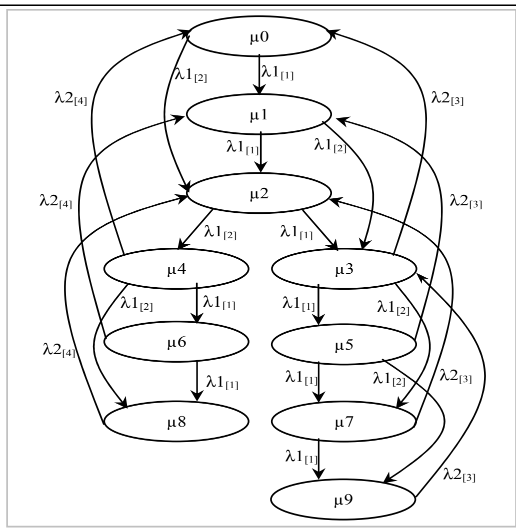

Fig. 5. The corresponding CTMC of the BDSPN.

$$\begin{bmatrix}
-2\lambda_1 & \lambda_1 & \lambda_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -2\lambda_1 & \lambda_1 & \lambda_1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -2\lambda_1 & \lambda_1 & \lambda_1 & 0 & 0 & 0 & 0 & 0 \\
\lambda_2 & 0 & 0 & -2\lambda_1-\lambda_2 & 0 & \lambda_1 & 0 & \lambda 1 & 0 & 0 \\
\lambda_2 & 0 & 0 & 0 & -2\lambda_1-\lambda_2 & 0 & \lambda_1 & 0 & \lambda_1 & 0 \\
0 & \lambda 2 & 0 & 0 & 0 & -2\lambda_1-\lambda_2 & 0 & \lambda_1 & 0 & \lambda_1 \\
0 & \lambda 2 & 0 & 0 & 0 & 0 & -\lambda_1-\lambda_2 & 0 & \lambda 1 & 0 \\
0 & 0 & \lambda_2 & 0 & 0 & 0 & 0 & -\lambda_1-\lambda_2 & 0 & \lambda_2 \\
0 & 0 & \lambda_2 & 0 & 0 & 0 & 0 & 0 & -\lambda_2 & 0 \\
0 & 0 & 0 & \lambda_2 & 0 & 0 & 0 & 0 & 0 & -\lambda_2
\end{bmatrix}$$

Fig. 6. The transition rate matrix A of the CTMC.

Table 1. Steady-state probabilities π

$$\pi0 = (\lambda_2)^2[\ 37.(\lambda_1)^3.\lambda_2+49(\lambda_1)^2.(\lambda_2)^2+24(\lambda_2)^3.\lambda_1+8(\lambda_1)^4+4(\lambda_2)^4] \div[164.(\lambda_1)^5.\lambda_2+326.(\lambda_1)^4.(\lambda_2)^2+361.(\lambda_1)^3.(\lambda_2)^3+32.(\lambda_1)^6+226.(\lambda_2)^4.(\lambda_1)^2+72.(\lambda_2)^5.\lambda_1+9.(\lambda_2)^6]$$

$$\pi1 = 2(\lambda2)2[\ 8(\lambda2)3.\lambda1+20(\lambda1)3.(\lambda2)+21(\lambda2)2.(\lambda1)2+4(\lambda1)4+(\lambda2)4] \div[164.(\lambda_1)^5.\lambda_2+326.(\lambda_1)^4.(\lambda_2)^2+361.(\lambda_1)^3.(\lambda_2)^3+32.(\lambda_1)^6+226.(\lambda_2)^4.(\lambda_1)^2+72.(\lambda_2)^5.\lambda_1+9.(\lambda_2)^6]$$

$$\pi2 = (2\lambda1+\lambda2).(\lambda1+\lambda2).(\lambda2)2\ [\ 3(\lambda2)2+20(\lambda1)2+15\lambda1\lambda2] \div[164.(\lambda_1)^5.\lambda_2+326.(\lambda_1)^4.(\lambda_2)^2+361.(\lambda_1)^3.(\lambda_2)^3+32.(\lambda_1)^6+226.(\lambda_2)^4.(\lambda_1)^2+72.(\lambda_2)^5.\lambda_1+9.(\lambda_2)^6]$$

$$\pi3 = \lambda1\lambda2.[\ 30(\lambda2)3.\lambda1+54(\lambda1)3.(\lambda2)+63(\lambda2)2.(\lambda1)2+5(\lambda2)4+16(\lambda1)4] \div \div[164.(\lambda_1)^5.\lambda_2+326.(\lambda_1)^4.(\lambda_2)^2+361.(\lambda_1)^3.(\lambda_2)^3+32.(\lambda_1)^6+226.(\lambda_2)^4.(\lambda_1)^2+72.(\lambda_2)^5.\lambda_1+9.(\lambda_2)^6]$$

$$\pi4 = (\lambda1+\lambda2).(\lambda2)2.\lambda1\ [\ 3(\lambda2)2+20(\lambda1)2+15\lambda1\lambda2] \div \div[164.(\lambda_1)^5.\lambda_2+326.(\lambda_1)^4.(\lambda_2)^2+361.(\lambda_1)^3.(\lambda_2)^3+32.(\lambda_1)^6+226.(\lambda_2)^4.(\lambda_1)^2+72.(\lambda_2)^5.\lambda_1+9.(\lambda_2)^6]$$

$$\pi5 = \lambda2.(\lambda1)2[5(\lambda2)3+23\lambda2(\lambda1)2+20(\lambda2)2.(\lambda1)+8(\lambda1)3] \div \div[164.(\lambda_1)^5.\lambda_2+326.(\lambda_1)^4.(\lambda_2)^2+361.(\lambda_1)^3.(\lambda_2)^3+32.(\lambda_1)^6+226.(\lambda_2)^4.(\lambda_1)^2+72.(\lambda_2)^5.\lambda_1+9.(\lambda_2)^6]$$

$$\pi6 = (\lambda2)2.(\lambda1)2\ [\ 3(\lambda2)2+20(\lambda1)2+15\lambda1\lambda2] \div \div[164.(\lambda_1)^5.\lambda_2+326.(\lambda_1)^4.(\lambda_2)^2+361.(\lambda_1)^3.(\lambda_2)^3+32.(\lambda_1)^6+226.(\lambda_2)^4.(\lambda_1)^2+72.(\lambda_2)^5.\lambda_1+9.(\lambda_2)^6]$$

$$\pi7 = [24(\lambda1)3+53(\lambda2)(\lambda1)2+\ 30(\lambda2)2(\lambda1)+5(\lambda2)3]\ .(\lambda1)2\ \lambda2 \div \div[164.(\lambda_1)^5.\lambda_2+326.(\lambda_1)^4.(\lambda_2)^2+361.(\lambda_1)^3.(\lambda_2)^3+32.(\lambda_1)^6+226.(\lambda_2)^4.(\lambda_1)^2+72.(\lambda_2)^5.\lambda_1+9.(\lambda_2)^6]$$

$$\pi8 = (\lambda1)2\ \lambda2\ [3(\lambda2)2+20(\lambda1)2+15\lambda2\lambda1].(2\lambda1+\lambda2) \div \div[164.(\lambda_1)^5.\lambda_2+326.(\lambda_1)^4.(\lambda_2)^2+361.(\lambda_1)^3.(\lambda_2)^3+32.(\lambda_1)^6+226.(\lambda_2)^4.(\lambda_1)^2+72.(\lambda_2)^5.\lambda_1+9.(\lambda_2)^6]$$

$$\pi9 = 2(\lambda1)3.[\ 38\lambda2(\lambda1)2+25(\lambda2)2.(\lambda1)+5(\lambda2)3+16(\lambda1)3] \div \div[164.(\lambda_1)^5.\lambda_2+326.(\lambda_1)^4.(\lambda_2)^2+361.(\lambda_1)^3.(\lambda_2)^3+32.(\lambda_1)^6+226.(\lambda_2)^4.(\lambda_1)^2+72.(\lambda_2)^5.\lambda_1+9.(\lambda_2)^6]$$

With the steady state probabilities, we can easily calculate some important performance indexes of the system, such as the average inventory level and the stockout rate of the system. The average inventory level of the system (the mean number of tokens in discrete place *p1*) can be obtained as:

$$\overline{\mu}(p1) = \sum_{i=0}^{9} \pi_i \times \mu(p1) = \frac{\lambda_2 \times [1126(\lambda_2)^2(\lambda_1)^3 + 569(\lambda_1)^4\lambda_2 + 88(\lambda_1)^5 + 932(\lambda_2)^3(\lambda_1)^2 + 341(\lambda_2)^4\lambda_1 + 46(\lambda_2)^5]}{164(\lambda_1)^5\lambda_2 + 326(\lambda_1)^4(\lambda_2)^2 + 361(\lambda_2)^3(\lambda_1)^3 + 32(\lambda_1)^6 + 226(\lambda_2)^4(\lambda_1)^2 + 72(\lambda_2)^5\lambda_1 + 9(\lambda_2)^6}$$

The stock-out rate of the system (the probability of the emptiness of place *p1*) is given by:

$$Prob(\mu(p1) = 0) = \pi_8 + \pi_9 = \frac{(\lambda_1)^2 \times [31(\lambda_2)^3(\lambda_1) + 100(\lambda_2)^2(\lambda_1)^2 + 116(\lambda_1)^3(\lambda_2) + 32(\lambda_1)^4 + 3(\lambda_2)^4]}{72(\lambda_2)^5(\lambda_1) + 9(\lambda_2)^6 + 226(\lambda_2)^4(\lambda_1)^2 + 361(\lambda_2)^3(\lambda_1)^3 + 326(\lambda_1)^4(\lambda_2)^2 + 164(\lambda_1)^5(\lambda_2) + 32(\lambda_1)^6}$$

Given λ1 = 1, the average inventory level and the stock-out rate as functions of λ2 are depicted in Fig. 7 and Fig. 8, respectively.

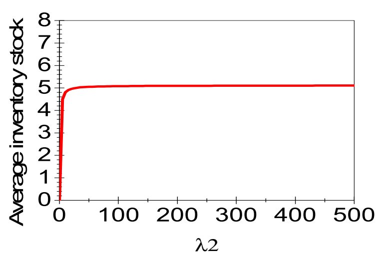

Fig. 7. Average inventory level of the stock.

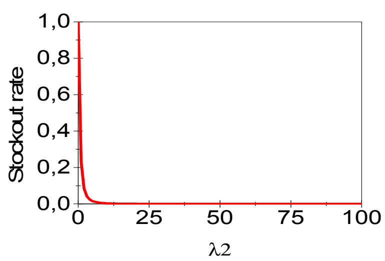

Fig. 8. Stock-out rate of the stock.

For other parameter values *s* and *S*, we can similarly calculate the performance indexes of the inventory system. The parameters can then be optimized by applying a local-search based meta-heuristic.

6. CONCLUSION

In this paper, a new stochastic Petri net model called batch deterministic and stochastic Petri nets (BDSPNs) is presented and its potential to the modelling and performance evaluation of inventory systems is illustrated. The most important advantages of the model are that it is capable of describing the synchronization of discrete and batch token flows appeared in batch discrete event systems and it keeps the simplicity of discrete Petri nets allowing us to extend existing methods for stochastic Petri nets to analyze it. We believe that the new model exhibits some important properties and is worth being further investigated in both theoretical and application aspects.

REFERENCES

Ajmone Marsan, M., et al., *Modelling with Generalized Stochastic Petri Nets*, Wiley Series in Parallel Computing, John Wiley and Sons, 1995.

Balduzzi, F., et al., "First-Order Hybrid Petri Net: A Model for Optimization and Control", *IEEE Trans. on Robotics and Automation*, vol. 16, no. 4, 2000.

Chen H., Amodeo L., and Chu F., Batch Deterministic and Stochastic Petri Nets: A Tool for Modeling and Performance Evaluation of Supply Chain, *Proc. of the 2002 IEEE Int. Conference on Robotics and Automation*, Washington D.C., May 2002, pp. 78-83.

Chen H. Amodeo L. Chu F., and Labadi K., Modeling and Performance Evaluation of Supply Chains Using Batch Deterministic and Stochastic Petri Nets, to appear in the April 2005 issue of *IEEE Trans. on Automation Science and Engineering*.

Furcas, R., et al., Hybrid Petri Net Modeling of Inventory Management Systems, *APII-JESA*, no. 35, 2001, pp. 417-434.

Lindemann, C., *Performance Modelling with Deterministic and Stochastic Petri Nets*, John Wiley and Sons, 1998.

Van der Aalst, W. M. P., "*Timed Colored Petri Nets and Their Application to Logistics*", PhD thesis, Technical University of Eindhoven, 1992.

Viswanadham N. Srinivasa Raghavan N. R., "Performance Analysis and Design of Supply Chains: A Petri Net Approach", *Journal of the Operational Research Society*, vol. 51, no. 10, 2000.

Zurawski R. and Zhou M., "Petri Nets and Industrial Applications: A tutorial" *IEEE Transactions on Industrial Electronics*, vol. 41, no. 6, 1994, pp. 567-583.