A simplified framework for stochastic workflow networks

Yajuan Li, Chuang Lin, Quan-Lin Li

Department of Computer Science and Technology, Tsinghua University, Beijing 100084, PR China
Department of Industrial Engineering, Tsinghua University, Beijing 100084, PR China

ARTICLE INFO

Article history:
Received 7 February 2007
Received in revised form 8 April 2008
Accepted 17 April 2008

Keywords:
Stochastic workflow networks
Crucial module
Simplification
PH distribution
Performance analysis

ABSTRACT

This paper presents a novel method to simplify stochastic workflow networks for their performance analysis under a unified computable framework. This method is based on two techniques: (1) module simplification, and (2) PH equivalence and PH approximation. In the first technique, simplified procedures for at least four crucial modules: sequential routing, parallel routing, selective routing and iterative routing are given, respectively; while in the second technique, the closure properties and the two-order approximation for the PH distributions are discussed. Using this method, we analyze several examples for the stochastic workflow networks and illustrate that performance evaluation of complicated stochastic workflow networks can be obtained by means of subsystems which are clearly constructed by some of the four structured modules. Numerical examples indicate that the method of this paper can tackle large-scale and complicated stochastic workflow networks with both effective approximation and low computational complexity.

1. Introduction

During the last three decades considerable attention has been paid to studying workflow. Workflow is an effort to automate and redesign the way that an organization tries to increase efficiency, reduce cost and create competitive advantages. Since the late 1980s, not only has workflow become a useful technology, but also it has been motivated by many practical applications including computer supported cooperative work (CSCW) systems, information systems, electronic commerce, manufacturing systems and communication networks. Hollingsworth [1] explained workflow as computerized facilitation or automation of a business process. Based on his interpretation, a large number of workflow products were developed such as FlowMark, MQSeries, FloWare, Action Workflow, Staffware, OPEN/workflow, InConcert, Visual Workflo and Lotus Notes (see, e.g. [2]). Also, some organizations made workflow management systems (WFMSs), which provide the ability to specify, execute, register and dynamically control such workflows. According to the workflow products and the WFMSs, some useful workflow patterns were abstracted, e.g., see [3]. Gamma et al. [4] catalogued 23 design patterns, which describe the smallest recurring interactions in object-oriented systems. It is worthwhile to note that several patterns constitute a basic routing, which always can be found in each workflow product. Besides, other patterns may be referred to as more advanced routings, which are difficult to be supported by workflow products. van der Aalst and van Hee [5] indicated that a workflow routing defines the order of performed tasks, and identified four basic routings including sequence, parallelism, choice and iteration. Kumar and Zhao [6] studied the dynamic routing of the WFMSs. Lin et al. [7] provided an exponential approximation for simplifying the four basic routings by using stochastic Petri nets (SPNs).

* Corresponding author. Tel.: +86 10 62783243; fax: +86 10 62794399.
E-mail address: liquanlin@tsinghua.edu.cn (Q.-L. Li).

0898-1221/$ – see front matter Crown Copyright © 2008 Published by Elsevier Ltd. All rights reserved.
Workflow should be well cared for its representation, reasoning and execution. Trajevski et al. [8] addressed the requirements of workflow specification. Useful specification languages have been proposed, for example, Opera Graphical Workflow Language (OGWL) by [9], State and Activity Charts by [10], Concurrent Transaction Logic (CTR) by [11], Transaction Datalog (TD) by [12] and Yet Another Workflow Language (YAWL) by [13]. Based on the 23 design patterns given by Gamma et al. [4], van der Aalst and ter Hofstede [14] recognized considerable differences in the expressive power of some workflow specification languages, and indicated that those languages can perform better to support the workflow patterns by using Petri nets (PNs). For a comprehensive introduction of PNs, readers may refer to a survey by [15] and a book by [16]. Reijers [17] reviewed workflow modeling and crucial techniques. Miller et al. [18] discussed simulation technology of workflow. van der Aalst [19] introduced PNs based analysis techniques for modeling checking of workflow. Adam et al. [20] concentrated on temporal logic analysis of workflow. A PN based language for workflow modeling, called Workflow net (WF-net), has attracted considerable interests. van der Aalst [21] showed two assumptions of a WF-net: (1) there is one input place and one output place, and (2) for each node there exists a directed path from the input place to the node and another directed path from the node to the output place, where the node is either a transition or a place. Except for the above two assumptions, he also indicated that the soundness should be needed. The soundness ensures that the process can terminate eventually, upon termination there is a single token in the output place while all the other places are empty, and more crucially, there should be no dead tasks. Dehnert and Rittgen [22] proposed the relaxed soundness.

Ludwig and Whittingham [23] and van Glabbeek and Stork [24] illustrated the importance and necessity of inter-organizational workflows, which consist of several cooperative organizations. To support inter-organizational workflow management, a specific research project, called CrossFlow, was started in 1989. Grefen et al. [25] gave an overview for the CrossFlow project. Subsequent papers have been published on modeling and verification of inter-organizational workflows, among which see van der Aalst [26], Yan and Wang [27], and Meng et al. [28].

Few methods were available to evaluate performance measures of the inter-organizational workflows. One of the major challenges results from the study of complicated stochastic systems, performance evaluation of which in general is intractable by means of ordinary Markovian analytic methods. To overcome difficulty due to a large state space of a Markov chain, simplification methods are always needed and have been proposed by some researchers, for example, state truncation by [29,30], state aggregation by [31–33], time scale decomposition by [34–36], fix-point iteration strategy by [37,38], and hierarchical decomposition by [39–41]. In this paper, we extend the inter-organizational workflow to a more general model: Stochastic workflow network, and then propose a novel method for computing performance measures of a stochastic workflow network under a unified approximate framework. This method consists of two steps: (1) using four crucial modules to simplify a more complicated network into a simple system, and (2) computing performance measures of the simple system, which are an effective approximation of these performance measures of the corresponding original network. Also, we show that such an approximation is more appropriate in the performance analysis of complicated stochastic workflow networks from a clear subsystem structure.

The main contributions of this paper are twofold. The first one is to provide four crucial modules, each of which can be simplified as one node of a workflow network under an equivalent setting of the PH distributions. For a comprehensive introduction of the PH distributions, readers may refer to a book [42] by Neuts. We believe that there must still exist other crucial modules with respect to simplification of complicated stochastic workflow networks, and the other modules can be similarly analyzed by the method for the four crucial modules proposed in this paper. Many practical examples indicate that a complicated stochastic workflow network in general can be decomposed into several key subsystems, thus such a decomposition technique is an important direction in the study of stochastic workflow networks. Specifically, the idea of subsystem decomposition has recently been developed by such as Stochastic Process Algebras (SPAs), e.g., see [43]. In SPAs, the four crucial modules can be regarded as basic operational elements. Each of these basic elements can be reduced by means of a lower-order PH distribution with a better approximation. Note that the PH approximation is able to widen the application fields of the SPAs. Therefore, the study of the four crucial modules becomes fundamental to our simplification techniques. The second contribution of this paper is to construct a simplification framework to deal with complicated stochastic workflow networks. This framework depends on the module based simplification with respect to all the key subsystems involved. Note that the PH distributions play an important role in the simplification framework, for example, the two-order PH distribution is frequently used in the simplification procedure, which makes the computational complexity low and the systemic structure simple. Therefore, the simplification method can be used to analyze large-scale and complicated systems more effectively than those in the literature. Some numerical examples indicate that the implementations of the simplification framework are convenient and efficient for stochastic workflow networks. It is worthwhile to note that Scarpa and Bobbio [44] applied PH distributions to SPNs and made an effort to reduce computer storage requirements via Kronecker algebra operators, but they do not consider structure simplification which can reduce computation complexity more effectively. He, Wu and Li [45] studied a longer production line by using the PH approximate method, the idea of which is further developed in this paper.

The remainder of this paper is organized as follows. Section 2 describes a stochastic workflow network, and also provides three simple and practical examples for a necessary interpretation. Section 3 analyzes four crucial modules and simplifies them under an equivalent or approximate setting based on the PH distributions. Section 4 presents a simplification framework in order to compute the performance measures of complicated stochastic workflow networks. Section 5 gives some numerical examples to indicate implementations of the simplification framework. The final section provides some concluding remarks.
2. Stochastic workflow networks

In this section, we describe a stochastic workflow network and provide four simple and practical examples for an interpretation purpose.

As mentioned in [26], a stochastic workflow network may be regarded as a set of loosely coupled business processes. Recall that a business process can be specified by a workflow net, which includes a source place, a sink place, and routings that define the order of task invocation. Therefore, a stochastic workflow network can be constructed by some workflow nets with interactions. In a stochastic workflow network, there may be more than one source place and/or sink place. Such models are suitable for representing business processes with different types of inputs that are treated in different ways and lead to different types of outputs. For an intuitive understanding, an example of the stochastic workflow network is depicted in Fig. 1.

We now use a SPN to describe a stochastic workflow network, which is expressed by a six-tuple \((S, T, F, R, I, O)\), where \(S\) and \(T\) are disjoint sets of places and transitions, \(F : (S \times T) \cup (T \times S) \rightarrow \mathbb{N}\) is a flow relation, \(R\) is a set of firing rates in which each element is the rate of the delay-time distribution for a transition, \(I\) is a set of source places where \(\forall i \in I, i^* = \emptyset\), and \(O\) is a set of sink places where \(\forall o \in O, o^* = \emptyset\). The symbols \(i^*\) and \(o^*\) denote the set of input nodes of \(i\) and the set of output nodes of \(o\), respectively.

For simplicity, we need to ensure the properness and the liveness of system specifications. Therefore, a stochastic workflow network is assumed to satisfy the following two key conditions: (1) there are no dead tasks in the network, and (2) each business process can terminate eventually.

In the rest of this section, we provide three simple and practical examples to help us understand the above meaning. The first example is a communication workflow consisting of two processes, as seen in Fig. 2 (a). Process 1 sends a message to Process 2, and cannot proceed until receiving an act from Process 2. Once Process 2 receives a message, Process 1 checks the
availability of the message, and then verifies the checksum and extracts header information concurrently. Such a model is useful for network protocols. The second example, as shown in Fig. 2 (b), represents a producer–consumer workflow. The producer and consumer are depicted as the left subnet and the right subnet, respectively. The consumer cannot start to work without the result sent by the producer. Such a model is useful for production and inventory systems. The third example is a resource sharing system depicted in Fig. 2 (c), where \( k \) tokens in a shared memory, denoted as place \( p \), represent that at most \( k \) processes may be reading concurrently, and while one process is writing, no other process can be reading or writing. Such a model can be used in computer operating systems.

As indicated by these concrete models, a stochastic workflow network always has a complicated structure, which makes performance evaluation difficult. It is obvious that performance analysis of such a stochastic workflow network is both theoretically necessary and practically important for system design, control and management. Therefore, a unified framework for simplifying complicated stochastic workflow networks is very useful for estimating the performance measures and will be analyzed in the next section in detail.

3. Module simplification

In this section, we analyze four crucial modules (or routings) and give their equivalent or approximate models with simplified structures in terms of the PH distributions. The four basic routings are: (1) sequential routing, (2) parallel routing, (3) selective routing and (4) iterative routing. Note that the four routings with exponential assumptions were first introduced by Aalst [21], while we here provide a more general discussion, which is useful for many practical systems.

Throughout this paper, \( \mathbf{e} \) denotes a column vector of ones, \( \mathbf{I} \) denotes an identity matrix, and \( \otimes \) and \( \oplus \) denote a Kronecker product operator and a Kronecker sum operator, respectively. Note that \( \mathbf{e} \) and \( \mathbf{I} \) can have the suitable sizes in the different places with a clear meaning.

3.1. Sequential routing

Sequential routing describes that the transitions fire sequentially. Sequential routing and its simplified node are shown in Fig. 3. If there are \( n \) sequential transitions \( t_1, t_2, \ldots, t_n \) in a sequential routing, and the delay time \( X_i \) of transition \( t_i \) is a PH-distributed random variable with irreducible representation \( (\alpha_i, T_i) \) of order \( k_i \), where \( \alpha_i \) is a probability vector of dimension \( k_i \) and \( T_i \) is a transition matrix of order \( k_i \) for \( 1 \leq i \leq n \), then the sequential routing can be simplified by a transition \( t \) with the total delay time \( X = \sum_{i=1}^{n} X_i \).

Let \( F \) and \( F_t \) be the probability distributions of \( X \) and \( X_t \), respectively. It is clear from \( X = \sum_{i=1}^{n} X_i \) that \( F = F_1 * F_2 * \cdots * F_n \), where \( G * H \) denotes the convolution of the two functions \( G \) and \( H \), i.e., \( G(x) * H(x) = \int_{0}^{\infty} G(x-u) \, dH(u) \). To compute the distribution \( F \), it is necessary to calculate the \( n \)-fold convolution iteratively. Thus, we provide the following algorithm.

**Algorithm 1.** Computation for Convolution of \( n \) PH distributions

**INPUT:** \( n \) independent PH distributions \( F_i \) with irreducible representation \( (\alpha_i, T_i) \) for \( 1 \leq i \leq n \).

**COMPUTATION:**

**Initialization:** Let \( F_{1,1} = F_1 \), which has the representation \( (\alpha_{1,1}, T_{1,1}) \), given by

\[
\alpha_{1,1} = \alpha_1 \quad \text{and} \quad T_{1,1} = T_1.
\]

**Repeat:** For \( j = 2 \) to \( n \), compute \( F_{1,j} = F_{1,j-1} * F_j \), which has the representation \( (\alpha_{1,j}, T_{1,j}) \), given by

\[
\begin{align*}
\alpha_{1,j} &= (\alpha_{1,j-1}, \quad (1-\alpha_{1,j-1})e \alpha_j), \\
T_{1,j} &= \begin{pmatrix} T_{1,j-1} & (\mathbf{-}T_{1,j-1}e) \alpha_j \\
\mathbf{0} & T_j \end{pmatrix}.
\end{align*}
\]

**Return:** \( \alpha = \alpha_{1,n} \) and \( T = T_{1,n} \).

**OUTPUT:** Convolution of the \( n \) PH distributions, i.e., \( F_{1,n} \) with irreducible representation \( (\alpha, T) \).

From the above procedure, we can see that the distribution \( F \) of \( X \) is of phase type with order \( (k_1 + k_2 + \cdots + k_n) \).

**Remark 1.** To understand **Algorithm 1** well, readers may refer to He, Wu and Li [45] for analyzing a large production line by means of the PH equivalence or approximation.
3.2. Parallel routing

Parallel routing means the transitions fire simultaneously or in any order. Parallel routing and its simplified node are shown in Fig. 4. If there are \(n\) parallel transitions \(t_1, t_2, \ldots, t_n\) in a parallel routing, and the delay time \(X_i\) of transition \(t_i\) is a PH-distributed random variable with irreducible representation \((\alpha_i, T_i)\) of order \(k_i\) for \(1 \leq i \leq n\), then the parallel routing can be simplified by a total transition \(t\) with the delay time \(X = \max (X_1, X_2, \ldots, X_n)\).

Let \(F\) and \(F_i\) be the distributions of \(X\) and \(X_i\), respectively. It is clear from \(X = \max (X_1, X_2, \ldots, X_n)\) that \(F = F_1F_2 \cdots F_n\). To compute the distribution \(F\), we provide the following algorithm.

**Algorithm 2.** Computation for Maximum of \(n\) PH Random Variables

**INPUT:** \(n\) independent PH distributions \(F_i\) with irreducible representation \((\alpha_i, T_i)\) for \(1 \leq i \leq n\).

**COMPUTATION:**

**Initialization:** Let \(F_{1,1} = F_1\), which has the representation \((\alpha_{1,1}, T_{1,1})\), given by

\[
\alpha_{1,1} = \alpha_1 \text{ and } T_{1,1} = T_1.
\]

**Repeat:** For \(j = 2\) to \(n\), compute \(F_{1,j} = F_{1,j-1} \times F_j\), which has the representation \((\alpha_{1,j}, T_{1,j})\), given by

\[
\alpha_{1,j} = (\alpha_{1,j-1} \otimes \alpha_j, (1 - \alpha_j e) \alpha_{1,j-1}, (1 - \alpha_{1,j-1} e) \alpha_j ),
\]

\[
T_{1,j} = \begin{pmatrix}
T_{1,j-1} \otimes T_j & \mathbf{I} \otimes (-T_j e) & (-T_{1,j-1} e) \otimes \mathbf{I}
\end{pmatrix}.
\]

**Return:** \(\alpha = \alpha_{1,n}\) and \(T = T_{1,n}\).

**OUTPUT:** Maximum of the \(n\) PH random variables, i.e., \(F_{1,n}\) with irreducible representation \((\alpha, T)\).

From the above procedure, we can see that the distribution \(F\) of \(X\) is of phase type with order \(\prod_{i=1}^{n} (k_i + 1) - 1\).

3.3. Selective routing

Selective routing means that the transitions fire mutually exclusively, i.e., all of them are enabled but exactly one of the alternatives is selected to fire. The selection among these transitions can be classified as two types: deterministic and nondeterministic. Nondeterministic selection indicates that the selection is completely left to the delay-time random variables associated with these transitions. In this case, the minimum of these random variables is selected. On the contrary, deterministic selection indicates that each transition is selected with some probability. Let us consider the two types of selective routings as follows.

3.3.1. Nondeterministic selection

Nondeterministic selective routing and its simplified node are shown in Fig. 5. If there are \(n\) selective transitions \(t_1, t_2, \ldots, t_n\) in a selective routing, and the delay time \(X_i\) of transition \(t_i\) is a PH-distributed random variable with irreducible representation \((\alpha_i, T_i)\) of order \(k_i\) for \(1 \leq i \leq n\), then the selective routing can be simplified by a total transition \(t\) with the delay time \(X = \min (X_1, X_2, \ldots, X_n)\). It is clear that \(F = 1 - \prod_{i=1}^{n} (1 - F_i)\), where \(F\) and \(F_i\) are the distributions of \(X\) and \(X_i\), respectively. The distribution \(F\) is also of phase type with irreducible representation \((\alpha, T)\) of order \((k_1 \times k_2 \times \cdots \times k_n)\), given by

\[
\alpha = \alpha_1 \otimes \alpha_2 \otimes \cdots \otimes \alpha_n,
\]

\[
T = T_1 \oplus T_2 \oplus \cdots \oplus T_n.
\]
3.3.2. Deterministic selection

Deterministic selective routing and its simplified node are shown in Fig. 6. If there are \( n \) selective transitions \( t_1, t_2, \ldots, t_n \) in a selective routing, the transition \( t_i \) can fire with probability \( p_i \) where \( 0 \leq p_i \leq 1 \) and \( \sum_{i=1}^{n} p_i = 1 \), and the delay-time distribution \( F_i \) of transition \( t_i \) is of phase type with irreducible representation \((\alpha_i, T_i)\) of order \( k_i \), for \( 1 \leq i \leq n \), then the selective routing can be simplified by a total transition \( t \) with the delay-time distribution \( F = \sum_{i=1}^{n} p_i F_i \). The distribution \( F \) is also of phase type with irreducible representation \((\alpha, T)\) of order \((k_1 + k_2 + \cdots + k_n)\), given by

\[
\alpha = (p_1\alpha_1, p_2\alpha_2, \ldots, p_n\alpha_n),
\]

\[
T = \begin{pmatrix}
T_1 & 0 & \cdots & 0 \\
0 & T_2 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & T_n
\end{pmatrix}.
\]

3.4. Iterative routing

The iterative routing may be regarded as the transitions fired repeatedly with a fixed probability. Intuitively, the iterative routing and its simplified node are shown in Fig. 7. If the iterative routing is composed of two transitions \( t_1 \) and \( t_2 \), the delay time \( X_i \) of transition \( t_i \) is a PH-distributed random variable with irreducible representation \((\alpha_i, T_i)\) of order \( k_i \) for \( 1 \leq i \leq 2 \), and the specific transition \( t_1 \) is selected to execute with probability \( p \) while the probability to leave the routing is \((1 - p)\), then the iterative routing can be simplified by a transition \( t \) with the total delay-time distribution \( F \), given by

\[
F = \sum_{j=0}^{\infty} (1 - p)^j p^j \left(F^{(j)}_1 * F^{((j+1))}_2\right),
\]

where \( H^{(j)} \) denotes the \( j \)-fold convolution of distribution \( H \).

Now, we show that \( F \) is of phase type and provide its irreducible representation under a simple form. Note that the convolution \( F_1 * F_2 \) is of phase type with irreducible representation \((\beta, S)\), where

\[
\beta = (\alpha_1, (1 - \alpha_1e) \alpha_2), \quad S = \begin{pmatrix}
T_1 & -T_1e\alpha_2 \\
0 & T_2
\end{pmatrix}.
\]
Table 1
Specific bounds of the first three moments

<table>
<thead>
<tr>
<th>Moment</th>
<th>Condition</th>
<th>Bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>(m_1)</td>
<td>(0 &lt; m_1 &lt; \infty)</td>
<td>(1.5m_2^2 \leq m_2)</td>
</tr>
<tr>
<td>(m_2)</td>
<td>(0.5 \leq c_2^2 \leq 1)</td>
<td>(3m_1^2 \left(3c_2^2 - 1 + \sqrt{2} (1 - c_2^2)^{\frac{3}{2}}\right) \leq m_2 \leq 6m_1^3c_2^2)</td>
</tr>
<tr>
<td>(m_3)</td>
<td>(1 &lt; c_2^2)</td>
<td>(\frac{3}{2}m_1^2(1 + c_2^2)^2 &lt; m_3 &lt; \infty)</td>
</tr>
</tbody>
</table>

Define

\[
H = \sum_{j=1}^{\infty} (1 - p)^{-1} F_1^{(j)} \ast F_2^{(j)},
\]

then \(H\) is of phase type with representation \((\beta, S - pSe\beta)\). It can be seen that \(F\) can be rewritten as

\[
F = (1 - p) F_2 + p (H \ast F_2).
\]

If \(H \ast F_2\) is expressed by \((v, \Gamma)\), where

\[
v = (\beta, (1 - \beta e) \alpha_2), \quad \Gamma = \begin{pmatrix} \alpha - pSe\beta & (1 - p) S^0 \alpha_2 \\ 0 & T_2 \end{pmatrix},
\]

then \(F\) is of phase type with irreducible representation \((\tau, \Psi)\), where

\[
\tau = ((1 - p) \alpha_2, pv), \quad \Psi = \begin{pmatrix} (1 - p) T_2 & 0 \\ 0 & p \Gamma \end{pmatrix}.
\]

On the other hand, the computation of the delay-time distribution \(F\) given in (1) can be given by matrix-exponential distributions in Section 4.2.

4. A simplification framework

In this section, we first give a simple overview for the moment fitting procedure when approximating a high-order PH distribution by using a two-order acyclic PH distribution (ACP(2)). It is worth noting that the new two-order PH distribution can have the same first three moments as that of the original high-order PH distribution. Then we apply the matrix-exponential distribution to compute the first three moments for a more general distribution with some complicated operations for several PH distributions, which in general may be very difficult in the ordinary setting. As such, we can also find a two-order PH approximation for the complicated distribution. Finally, we provide an effective algorithmic framework to simplify complicated stochastic workflow networks.

4.1. PH approximation

The fitting procedure, given in [46], provides mapping from a high-order PH distribution into the representation of a new two-order PH distribution with the first three moments matched exactly. To show this, it is necessary to recall the fitting procedure simply.

A PH distribution is expressed by

\[
F(t) = 1 - \alpha e^{T_2}, \quad \text{for } t \geq 0,
\]

where \(\alpha\) is a \(k\)-dimensional probability vector, and \(T\) is a nonsingular matrix of size \(k\), where \(T_{ii} < 0\) for \(1 \leq i \leq k\), \(T_{ij} \geq 0\) for \(i \neq j\), and \(Te \preceq 0\). The \(n\)th moment of the PH distribution is given by

\[
m_n = E[X^n] = n! \alpha (T)^{-n} e, \quad n \geq 1.
\]

Now, we take an ACP(2) distribution with the canonical representation \((\tilde{\alpha}, \tilde{T})\), given by

\[
\tilde{\alpha} = (\mu, 1 - \mu) \quad \text{and} \quad \tilde{T} = \begin{pmatrix} -\lambda_1 & \lambda_1 \\ 0 & \lambda_2 \end{pmatrix},
\]

where \(0 \leq \mu \leq 1\) and \(0 < \lambda_1 \leq \lambda_2\).

Our final purpose is to give the three unknown parameters \(\mu, \lambda_1\) and \(\lambda_2\) of ACP(2) by means of the first three moments \(m_1, m_2\) and \(m_3\). We denote by \(c_2^2\) the squared coefficient of variation given by \(c_2^2 = m_2/m_1^2 - 1\). Let \(c = 3m_2^2 - 2m_1 m_3\), \(d = 2m_2^2 - m_2\), \(b = 3m_1 m_2 - m_3\), and \(a = b^2 - 6cd\), if these moments \(m_1, m_2\) and \(m_3\) satisfy the corresponding specific bounds given in Table 1.
Based on this, the parameters of ACPH(2) can be computed according to the following four cases.

(1) If $c > 0$, then
$$
\mu = \frac{-b + 6m_1 d + \sqrt{a}}{b + \sqrt{a}}, \quad \lambda_1 = \frac{b - \sqrt{a}}{c}, \quad \lambda_2 = \frac{b + \sqrt{a}}{c}.
$$

(2) If $c < 0$, then
$$
\mu = \frac{b - 6m_1 d + \sqrt{a}}{-b + \sqrt{a}}, \quad \lambda_1 = \frac{b + \sqrt{a}}{c}, \quad \lambda_2 = \frac{b - \sqrt{a}}{c}.
$$

(3) If $c = 0$, then
$$
\mu = 0, \quad \lambda_1 \text{ is irrelevant}, \quad \lambda_2 = \frac{1}{m_1}.
$$

(4) If the first three moments do not satisfy the conditions given in Table 1, then we may analyze three examples:

(4.1) If $m_2 < 1.5m_1^2$, then we take $m_2 = 1.5m_1^2$,

(4.2) if $0.5 \leq c_X^2 \leq 1$, $6m_1^2 c_X^2 < m_3 < 3m_1^2 \left(3c_X^2 - 1 + \sqrt{2}(1 - c_X^2)\right)$, then we take $m_2 = 3m_1^2 \left(3c_X^2 - 1 + \sqrt{2}(1 - c_X^2)\right)$ and $m_3 = 6m_1^2 c_X^2$.

(4.3) if $c_X^2 > 1$ and $m_3 \leq \frac{3}{2}m_1^2 \left(1 + c_X^2\right)$, then we take $m_3 = \frac{3}{2}m_1^2 \left(1 + c_X^2\right)$.

From the above discussion, we can always construct a two-order PH distribution to approximate an arbitrary general distribution with the same first three moments. To compute the first three moments of a general distribution, it may be convenient to use the matrix-exponential distribution.

### 4.2. Matrix-exponential distribution

In a stochastic workflow network, we always encounter a complicated operation of several PH distributions, as illustrated in Section 3 for the delay-time distribution. It may be convenient to use the matrix-exponential distribution to compute the first three moments of the delay-time distribution.

As defined in [47], a matrix-exponential distribution $B$ is given by

$$
B(x) = 1 + \beta e^{\lambda t},
$$

where $\beta$ and $t$ are a row vector and a column vector respectively, $L$ is a square matrix, $\beta L^{-1} t = -1$ and $\beta t \geq 0$. We refer to the triple $(\beta, L, t)$ as the representation of the distribution $B$. It is clear that a PH distribution can be regarded as a special matrix-exponential distribution.

The Laplace–Stieltjes transform of the matrix-exponential distribution is given by

$$
\hat{b}(s) = \beta (sI - L)^{-1} t,
$$

whose polynomial representation is specified as

$$
\hat{b}(s) = \frac{b_n's^{n-1} + b'_{n-1}s^{n-2} + \ldots + b'_{2}s + b'_1}{s^n + b'_n's^{n-1} + \ldots + b'_{n-1}s + b''_n},
$$

where coefficients $b'_i$ and $b''_i$ are constants for $1 \leq i, j \leq n$. According to Proposition 2.3 of [47], when $b'_1 = b''_n$, the distribution has the representation $(\alpha, V, p)$, where

$$
\alpha = (b'_1, b'_2, \ldots, b'_n), \quad p = (0, 0, \ldots, 0, 1)^\top
$$

and

$$
V = \begin{pmatrix}
0 & 1 & 0 & \cdots & 0 \\
0 & 0 & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & 1 \\
-b''_n & -b''_{n-1} & -b''_{n-2} & \cdots & -b''_1
\end{pmatrix},
$$

where the superscript $T$ denotes matrix transpose. It is worth noting that the $n$th moment of the distribution $B$ is given by $m_n = n!\alpha (-V)^{-n-1} p$ for $n \geq 1$. 

As an example, we illustrate how to use the matrix-exponential distribution to compute the first three moments of the delay-time distribution for the iterative routing given in Section 3.4. We assume that the Laplace–Stieltjes transforms of the two distributions $F_1$ and $F_2$ are respectively given by

$$\hat{f}_1(s) = \frac{c'_k s^{k-1} + c'_{k-1} s^{k-2} + \cdots + c'_1 s + c'_0}{s^k + c'_1 s^{k-1} + \cdots + c''_{k-1} s + c''_0},$$

and

$$\hat{f}_2(s) = \frac{d'_k s^{k-1} + d'_{k-1} s^{k-2} + \cdots + d'_1 s + d'_0}{s^k + d'_1 s^{k-1} + \cdots + d''_{k-1} s + d''_0}.$$  

(9)

(10)

It is easy to see from (1) that Laplace–Stieltjes transform $\hat{f}(s)$ of the delay-time distribution $F$ is given by

$$\hat{f}(s) = \sum_{j=0}^{\infty} (1-p) p^j [\hat{f}_1(s)]^j [\hat{f}_2(s)]^{j+1}.$$  

Thus, we obtain

$$\hat{f}(s) = \frac{(1-p) \hat{f}_2(s)}{1 - p \hat{f}_1(s) \hat{f}_2(s)}.$$  

By substituting (9) and (10) into the function $\hat{f}(s)$, we can obtain

$$\hat{f}(s) = \frac{a'_{k_1+k_2} s^{k_1+k_2-1} + a'_{k_1+k_2-1} s^{k_1+k_2-2} + \cdots + a' s + a'_0}{s^{k_1+k_2} + a''_{k_1+k_2-1} s^{k_1+k_2-2} + \cdots + a'' s + a''_0},$$

which yields the representation $(\gamma, W, \mathbf{q})$ of the matrix-exponential distribution corresponding to (11), where

$$\gamma = (a'_1, a'_2, a'_3, \ldots, a'_{k_1+k_2})^T, \quad \mathbf{q} = (0, 0, \ldots, 0, 1)^T$$

and

$$W = \begin{pmatrix}
0 & 1 & 0 & \cdots & 0 \\
0 & 0 & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & 1 \\
-a''_{k_1+k_2} & -a''_{k_1+k_2-1} & -a''_{k_1+k_2-2} & \cdots & -a''_0
\end{pmatrix}.$$  

Therefore, the $n$th moment of the delay-time distribution $F$ is given by

$$m_n = n! \gamma (-W)^{-n-1} \mathbf{q}.$$  

It is worth noting that expression for the $n$th moment of the delay-time distribution $F$ may be rather complicated by using the ordinary method for operations of the PH distributions, while the Laplace–Stieltjes transform can simplify such operations by means of matrix-exponential distribution, as seen in the above example.

4.3. A simplification algorithm

Based on the PH approximation and the matrix-exponential distributions described above, we now provide an effective algorithm to simplify a complicated stochastic workflow network.

Algorithm 3. Simplification of a stochastic workflow network

**INPUT:** A stochastic workflow network: structure and parameters.

**PROCEDURE:**

Step 1: Recognize crucial modules of the network in terms of the results given in Section 3.

Step 2: For each module, if orders of delay-time distributions associated with transitions are very high, go to Step 3; otherwise, simplify the routing to one transition with the same delay-time distribution.

Step 3: Apply the two-order PH approximation and the matrix-exponential distributions to reduce the PH order of the delay-time distribution associated with each transition.

Step 4: Repeat the first two steps until the final model doesn’t involve any basic module. When all the basic modules are simplified, simplifying the stochastic workflow network is finished. Note that the final simplified model provides the appropriate performance measures to that of the original one.

**OUTPUT:** The simplified model.
The steps in Algorithm 3 for simplifying a stochastic workflow network are shown in Fig. 8.

The first advantage of Algorithm 3, compared with ordinary Markovian analysis methods, is that the computational complexity is reduced greatly so that it can be used to deal with some complicated stochastic workflow networks. The second advantage is that the PH distribution and the PH approximation are extensively applied in our study. Note that the PH distributions possess many convenient properties, thus the simplification procedure can be performed very well. Since a general probability distribution defined on the interval \([0, \infty)\) can be approximated by a two-order PH distribution with the same first three moments, Algorithm 3 can be applied to deal with more stochastic workflow networks under a unified computational framework.

Remark 2. When a stochastic system is complicated, a decomposition method under the same mean, corresponding to the exponential approximation, is always used, e.g., see Gershwin [41]. In this paper, we use more useful information with the first three moments, corresponding to the two-order PH approximation, which improves the approximate level, e.g., see He, Wu and Li [45]. In general, it is difficult for comparing the approximate and exact solutions for performance analysis of a complicated system, because an exact solution does not exist possibly. To our best knowledge, the approximate method with the first three moments given in this paper is the best one in the literature.

5. Numerical examples

In this section, we give four numerical examples to illustrate how to use the algorithm presented in Section 4.3. Specifically, we provide some useful observations with necessary interpretations on effective simplification of two stochastic workflow networks.

In the first example, we consider a simple stochastic workflow network with one source place and one sink place, which is shown in Fig. 9 (1), where \(t_i\) denotes a timed transition whose delay-time distribution is of phase type with irreducible representation \((\alpha_i, T_i)\) for \(1 \leq i \leq 10\), and the branch probabilities \(\beta_1\) and \(\beta_2\) are 0.6 and 0.8, respectively. We further take the following parameters:

- \(\alpha_1 = (0.1, 0.6, 0.3)\), \(\alpha_2 = (0.3, 0.2, 0.5)\), \(\alpha_3 = (0.2, 0.1, 0.7)\), \(\alpha_4 = (0.4, 0.2, 0.4)\),
- \(\alpha_5 = (0.8, 0.1, 0.1)\), \(\alpha_6 = (0.4, 0.3, 0.3)\), \(\alpha_7 = (0.2, 0.6, 0.2)\),
- \(\alpha_8 = (0.4, 0.4, 0.2)\), \(\alpha_9 = (0.9, 0.1, 0)\), \(\alpha_{10} = (0.5, 0.1, 0.4)\),

- \(T_1 = \begin{pmatrix} -0.4 & 0.3 & 0.1 \\ 0.3 & -0.6 & 0.2 \\ 0.5 & 0.2 & -0.8 \end{pmatrix}\), \(T_2 = \begin{pmatrix} -0.3 & 0.1 & 0.2 \\ 0 & -0.2 & 0.1 \\ 0.4 & 0 & -0.5 \end{pmatrix}\),
- \(T_3 = \begin{pmatrix} -0.2 & 0 & 0.1 \\ 0.1 & -0.4 & 0.2 \\ 0.3 & 0.1 & -0.7 \end{pmatrix}\), \(T_4 = \begin{pmatrix} -0.5 & 0.2 & 0.2 \\ 0.1 & -0.3 & 0.1 \\ 0 & 0.3 & -0.4 \end{pmatrix}\).
Fig. 9. Simplification of a stochastic workflow network.

\[ T_5 = \begin{pmatrix} -0.3 & 0.2 & 0 \\ 0 & -0.5 & 0.4 \\ 0.2 & 0.3 & -0.6 \end{pmatrix}, \quad T_6 = \begin{pmatrix} -0.6 & 0.4 & 0.2 \\ 0.2 & -0.8 & 0.2 \\ 0.1 & 0.2 & -0.4 \end{pmatrix}, \]

\[ T_7 = \begin{pmatrix} -0.1 & 0.1 & 0 \\ 0.4 & -0.7 & 0.2 \\ 0.3 & 0.3 & -0.9 \end{pmatrix}, \quad T_8 = \begin{pmatrix} -0.9 & 0.4 & 0.3 \\ 0.5 & -0.8 & 0.2 \\ 0.2 & 0.1 & -0.5 \end{pmatrix}, \]

\[ T_9 = \begin{pmatrix} -0.5 & 0.2 & 0.3 \\ 0.2 & -0.4 & 0.1 \\ 0 & 0.1 & -0.2 \end{pmatrix}, \quad T_{10} = \begin{pmatrix} -0.3 & 0.1 & 0.2 \\ 0.5 & -0.6 & 0 \\ 0.1 & 0.2 & -0.4 \end{pmatrix}. \]

Using the algorithm given in Section 4.3, the stochastic workflow network can be simplified to the final transition \( t_{1,10} \) with three simplifying steps, as indicated in Fig. 9. It is easy to check by using the basic modules given in Section 3 that the first three moments of \( t_{1,10} \) are given by

\[ m_1 = 1.1217e + 002, \quad m_2 = 2.0658e + 004, \quad m_3 = 5.4005e + 006. \]  \hspace{1cm} (12)

We now use the two-order PH approximations for those high-order PH distributions involved in the procedure for studying \( t_{1,10} \). To check the accuracy of the algorithm for computing the delay-time distribution, we use the PH equivalence and the two-order PH approximation in our simplification. In the final step, the first three moments of \( t_{1,10} \) are approximately given by

\[ m_1 = 1.1216e + 002, \quad m_2 = 2.0491e + 004, \quad m_3 = 5.5321e + 006. \]  \hspace{1cm} (13)

By comparing (12) with (13), it can be seen that the algorithm has a higher precision.

To give more data samples about comparing the exact results and the approximations, we vary the initial probability distributions or the transition matrices of these PH distributions. The chosen model is the same as that in the first example. We provide two different studies. Firstly, we take \( \alpha_1 = (x, 0.7 - x, 0.3) \) for \( 0 \leq x \leq 0.7 \), while all the other vectors \( \alpha_i \) for \( 2 \leq i \leq 10 \) and all the matrices \( T_j \) for \( 1 \leq j \leq 10 \) are the same as those in the first example. Fig. 10 compares the first three moments of \( t_{1,10} \) obtained by using the ordinary exact method and by using our algorithm. It can be seen that the first three moments of the delay time all increase as \( x \) grows.
Secondly, we vary the transition probability matrices and only take a different matrix as follows:

\[
T_1 = \begin{pmatrix}
-0.4 & x & 0.1 \\
0.3 & -0.6 & 0.2 \\
0.5 & 0.2 & -0.8
\end{pmatrix},
\]

where \(0 \leq x \leq 0.3\). Fig. 11 compares the first three moments of \(t_{1,10}\) obtained by using the exact method and by using our algorithm. It can be seen that the first three moments of the system delay time all increase as \(x\) grows.

In the second example, we consider a stochastic workflow network for describing a practical supply system with three business partners: A producer and two suppliers. The producer sends some orders for item \(a\) and item \(b\) to supplier 1 and supplier 2, respectively. Once supplier 1 receives an order for item \(a\), it will start a two-stage production procedure. In the first stage, supplier 1 needs to use four parallel production lines to execute certain tasks synchronously. In the second stage, supplier 1 selects one of the three workers to continue the production. Once supplier 2 receives an order for item \(b\), it will also start a two-stage production procedure. In the first stage, supplier 2 checks the order with two parallel tasks: Checking whether the order is available, and checking whether all the accessories for producing item \(b\) are on hand. In the second stage, supplier 2 continues the production by using three parallel production lines. Finally, items \(a\) and \(b\) are delivered to the producer to be assembled into a final product. Fig. 12 depicts the interaction between the three partners. Fig. 13 (1) specifies the corresponding model of the stochastic workflow network, where the branch probabilities \(\beta_0, \beta_1, \beta_2\) and \(\beta_3\) are 0.2, 0.3, 0.5 and 0.4, respectively, the firing delay of each transition is exponentially distributed and the firing rate of each transition is given in Table 2. By using our algorithm, the stochastic workflow network can be simplified through four steps, as indicated in Fig. 13. In this example, we largely reduce the order of the final PH distribution although the corresponding exact PH distribution has a higher order over 100. It is easy to calculate the approximate processing delay times of the three partners by using the computational methods given in Section 3. These approximate processing delay times are given in the last three lines of Table 3 and are specified by their first three moments. Using Stochastic Petri Net Package (SPNP) [48], we can get the accurate average processing delay time for each partner as shown in the first line of Table 3. It is worth noting that the difference between the approximate solution and the accurate solution is very small, which can demonstrate that our algorithm achieves high accuracy to some extent. Also, the first three moments given by our algorithm can specify the objective quantities with more information than that by only the mean throughout the ordinary exponential approximate algorithms, e.g., see a decomposition method by Gershwin [41]. Furthermore, our algorithm largely reduces the computer storage requirement which grows almost linearly with the state-space size of the given model instead of exponentially.
In the remainder of this section, we give a direct simplification for several stochastic workflow networks without concrete PH computations, which in general are not difficult in terms of the computational framework given in Sections 3 and 4. The graphic simplification is useful and helpful for us to understand the structured approximation of a stochastic workflow network. In Fig. 14, we provide the concrete steps of the graphic simplification for the three stochastic workflow networks given in Fig. 2.
6. Concluding remarks

In this paper, a novel simplified framework is proposed for studying complicated stochastic workflow networks. For four crucial modules: sequential routing, parallel routing, selective routing and iterative routing, we discuss their two-order PH equivalence and approximation under keeping the same first three moments. Numerical implementations indicate that this method is effective for dealing with complicated stochastic workflow networks. Unlike the ordinary methods suffering from rapid explosion of the state space, this method can simplify a stochastic workflow network into a simpler model module by module. As such, the solving structure of a complicated stochastic workflow network can be controlled within an allowable and tractable range. It is worth mentioning that this method can be expanded to arbitrary SPNs as long as we can recognize these basic modules. As indicated in Li, Wang and Zhou [49] and Li [50], there are still a number of directions for potential future research, for example,

(1) providing other key modules, such as advanced synchronization patterns and cancelation patterns, and giving the corresponding simplifications,
(2) providing more effective algorithms with computational complexity analysis for stochastic workflow networks, and
(3) applying the simplified computation for performance evaluation of stochastic workflow networks to some practical areas such as finance, business, communication networks and manufacturing systems.

Acknowledgments

The authors thank three anonymous referees for their detailed comments and helpful suggestions. The work of Y. Li and C. Lin was supported in part by the National Natural Science Foundation of China under Grant No. 60673187 and the National Grand Fundamental Research 973 Program of China under Grant No. 2006CB708301. The work of Q.L. Li was supported in part by the National Natural Science Foundation of China under Grant No. 10671107 and No. 60736028, and the National Grand Fundamental Research 973 Program of China under Grant No. 2006CB805901.
Fig. 14. Structured simplification of three stochastic workflow networks.

References