Model Check Stochastic Supply Chains

Li Tan
School of Electrical Engineering
and Computer Science
Washington State University
Richland, WA 99354, USA
Email: litan@tricity.wsu.edu

Shenghan Xu
College of Business and Economics
University of Idaho
Moscow, ID 83843, USA
Email: shenghan@uidaho.edu

***Abstract*—Supply chain [2], [6] is an important component of business operations. Understanding its stochastic behaviors is the key to risk analysis and performance evaluation in supply chain design and management. We propose a novel computational framework for modeling and analyzing the stochastic behaviors of a supply chain. The framework is based on probabilistic model checking, a formal verification technique for analyzing stochastic systems. Our approach is two-fold: first, we developed Stochastic Supply Chain Model (SMF), a formal framework for modeling stochastic supply chains based on Extended Markov Decision Process (EMDP); second, we proposed a model-checking-based formal technique to automate the analysis of a stochastic supply chain. Our model-checking-based approach leverages benefits of recent advances in symbolic probabilistic model checking to improve the efficiency and scalability of decision procedures. Using the temporal logic PCTL [1] and the symbolic probabilistic model checker PRISM [4], we are able to express and check complicate temporal and stochastic properties on supply chains. Finally, we demonstrate the capability of our model-checking-based approach by testing it on a variety of stochastic supply chain models.**

I. Introduction

In today's global economy, companies are increasingly relying on oversea suppliers to cater differentiated customer requirements and gain competitive edge in the market. This dependency on oversea suppliers also increase uncertainties and risks to already complex global supply-chain design and management. Recent recalls on toys and food in the United States serve as a weak-up call for risks in global supply chains: if not adequately being evaluated and addressed, catastrophic events like these could paralyze a supply chain and cause substantial financial loss to a company. Because the size of a global supply chain and the complexity arising from interactions among different units render traditional modeling and analysis methods intractable, evaluating and managing risks in a global supply chain presents a daunting challenge in a company's logistic planning. Risks root from uncertainty and stochastic elements in supply chains, and understanding stochastic behaviors of a supply chain is the key to assessing and managing risks in supply chains.

We propose a novel computational framework for modeling and analyzing stochastic behaviors of a supply chain. The framework is based on probabilistic model checking [1], a formal verification technique developed in Computer Science for analyzing stochastic systems. Recent advances [4] in symbolic probabilistic model checking have drastically improved its efficiency and scalability. In past several years the application of probability model checking has been extended from compute-based systems to a wide range of other subjects such as biological pathway [3]. A typical supply chain contains a large number of stochastic elements in manufacturing and transportation processes. The scale and complexity of the problem make it an ideal candidate for applying probabilistic model checking. Yet due to the interdisciplinary nature of this research, to the best of our knowledge this is the first reported effort to apply probabilistic model checking to the area of risk analysis in supply chain management.

Our approach is two-fold: first, we developed a formal framework for modeling a stochastic supply chain. The framework, Stochastic Merchandise Flow Model (SMF) is based on an extension of Markov Decision Process. The framework provides a rigorous approach for modeling dynamics of stochastic supply chains and it removes ambiguity in a supply chain design. SMF enables the composition of both stochastic elements, for example, warehouses that may fail, and nondeterministic elements, for example, routing decisions, in a single framework. SMF also provides the foundation for the formal analysis of a stochastic supply chain; second, we established a procedure for applying probabilistic model checking to stochastic supply chains. A probabilistic model checker such as PRISM [4] checks a stochastic system against a property encoded in a temporal logic. Use of a temporal logic allows us to customize and check complicate stochastic properties which are not supported by existing domain-specific and ad-hoc supply-chain risk analysis tools.

To assess the capability of our approach, we apply it to a variety of supply chain designs and report our experimental results. We use the symbolic probabilistic model checker PRISM in our experiments. PRISM supports the analysis of Markov Decision Process and uses a bounded variant of the probabilistic temporal logic PCTL to encode properties. We will discuss our framework in context of PRISM. The rest of the paper is organized as follows: Section II provides a brief introduction to probabilistic model checking and the symbolic probabilistic model checker PRISM. Section III discusses SMF, a formal framework we introduced to model stochastic supply chains. Section IV introduces the procedure we proposed for using probabilistic model checking to analyze stochastic supply chains. In Section V we discuss our experimental results. Finally Section VI concludes the paper.

II. Probabilistic Model Checking with PRISM

Model checking is a formal verification technique that algorithmically checks a dynamic system against a temporal property encoded in a temporal logic. Probabilistic model checking extends classical model checking techniques with the ability of reasoning stochastic behaviors of a system. In addition to simple "yes/no" answer, a probabilistic model checker also returns the probability with which a property may hold on a system. Probabilistic model checking has been used to analyze performance and reliability issues for a variety of subjects in Computer Science and other fields [5] where computational assessment of stochastic behaviors is the key to the answers. A recent advance in probabilistic model checking is the development of sophisticated symbolic techniques that greatly improves the scalability and efficiency of decision procedures. Since we use the symbolic probabilistic model checker PRISM in our experiment, We will discuss probabilistic model checking in context of PRISM.

Probabilistic model checking starts with the formal modeling of stochastic systems. Stochastic supply chains we are studying have both probabilistic elements and nondeterministic elements. A classical method for modeling a stochastic system with nondeterministic behaviors is Markov Decision Process (c.f. [7]). Definition 1 defines EMDP, an extension of Markov Decision Process that we use as the mathematical foundation for our modeling framework.

Definition 1 (Extended Markov Decision Process): An Extended Markov Decision Process (EMDP) is a tuple $\langle \mathbf{V}, \mathbf{D}, \mathbf{w}^0, \mathcal{D}, P, N\rangle$, where,

1) $\mathbf{V} = \langle v_1, \cdots, v_k\rangle$ is a vector of internal variables and its domain is $\mathbf{W} = W_1 \times W_2 \times \cdots \times W_k$;
2) $\mathbf{D} = \langle d_1, \cdots, d_l\rangle$ is a vector of external variables and its domain is $\mathbf{Z} = Z_1 \times Z_2 \times \cdots \times Z_k$;
3) $\mathbf{w}^0 = \langle w_1^0, \cdots, w_k^0\rangle$ is the initial valuation of $\mathbf{W}$.
4) $P : \mathbf{W} \times \mathbf{Z} \rightarrow 2^{\mathbf{W}\times(0,1]}$ is the probabilistic transition function, such that for every $\mathbf{w} \in dom(P)$, $\sum_{\langle \mathbf{w}', p\rangle \in P(\mathbf{w},\mathbf{z})} p = 1$.
5) $N : \mathbf{W} \times \mathbf{Z} \rightarrow 2^{\mathbf{W}}$ is the nondeterministic transition function, such that $dom(N) \cap dom(P) = \emptyset$.

□

It shall be noted that EMDP is not more expressive than Markov Decision Process. Nevertheless, it does bring several benefits that makes stochastic system modeling more effective and concise: first, EMDP uses a vector of variables to encode the state space of a Markov Decision Process. This allows us to represent sets of states and transitions more efficiently. For example, one may use a predicate $(v_i > a) \wedge (v_j < b)$ to represent the set of all the states such that $v_i > a$ and $v_j < b$. Second, we make a clear distinction between probabilistic transitions and nondeterministic transitions by representing them separately using two different transition functions. In context of stochastic supply chains, probabilistic transitions represent risks in operations, for example, the probability of failure of a warehouse, and nondeterministic transitions represent options in scheduling such as how to route merchandise flows. As an extreme case, when an EMDP has only probabilistic transitions, it is reduced to Discrete Time Markov Process, in which system behaviors are left to the probability, as shown in Figure 3. On the other hand, when an EMDP has only a nondeterministic transition function, it is reduced to a nondeterministic finite automaton, as shown in Figure 4.

$$f ::= A \mid \neg f \mid f \wedge f \mid [P_{\bowtie p}]\phi$$
$$\phi ::= \mathsf{X} f \mid f \mathsf{U} f \mid f \mathsf{R} f$$

where $A \in \mathcal{A}$ is an atomic proposition, $p \in [0, 1]$, and $\bowtie \in \{\leq, \geq, <, >\}$.

Fig. 1. The syntax of PCTL [1], [4]

We use the temporal logic PCTL [1] to express properties we want to check on a stochastic supply chain model. PCTL is a probabilistic extension of Computation Tree Logic (CTL). The syntax of PCTL is given in Figure 1.

PCTL has two types of formulae: state formulae and path formulae. Semantically a state formula represents a set of states, and a path formula represents a set of paths. We let variables $\phi, \psi, \cdots$ and $f, g, \cdots$ range over path formulae and state formulae, respectively. PCTL is a propositional logic and it is built upon atomic propositions. An atomic proposition A represents a set of states by its semantic definition. Particularly, the atomic propositions $\mathbf{T}$ and $\mathbf{F}$ stand for the set of all the states of a stochastic system and the empty set, respectively.

PCTL uses a set of path operators *next* (X), *until* (U), and *release* R to express temporal patterns. A path formula $\mathsf{X}f$ holds on a path $s_1 s_2 \cdots$ if s_2 satisfies f. $f_2 \mathsf{R} f_1$ holds on a path ρ if f_1 holds for every state on ρ unless a state s_i satisfying f_2 "releases" such obligation, in which case f_1 does not have to hold for states after s_i. $f_1 \mathsf{U} f_2$ holds on a path ρ if f_1 holds for every state "until" a state s_i satisfying f_2, after which f_1 may or may not hold. Note that a subtlety is that f_2 eventually holds at some state on β in $f_1 \mathsf{U} f_2$ but not necessarily so in $f_2 \mathsf{R} f_1$. We also use two additional path operators *always* ($\mathbf{G}$) and *eventually* $\mathbf{F}$. $\mathbf{G}f$ and $\mathbf{F}f$ stand for $\mathbf{F}\mathsf{R}f$ and $\mathbf{T}\mathsf{U}f$, respectively.

PCTL extends CTL with the probabilistic operator P, which attaches a probability to a path formula. For example, $P_{>0.5}\phi$ is true for a state s if the probability that ϕ holds on the paths from s is greater than 0.5. PRISM also allows a user to query the probability associated with a path formula. Model checking $P_{=?}\phi$ on a state s will yield the probability that ϕ holds on the paths from s. PRISM also supports a bounded version of path operators. For example, $P_{=?}(F^{\leq 6} f)$ queries the probability that a state satisfying f can be reached within 6 steps from the current state. Interested readers may refer to [1] for a detailed discussion on the semantics of PCTL.

Since a Markov Decision Process may have nondeterministic elements, the probability associated with a path formula

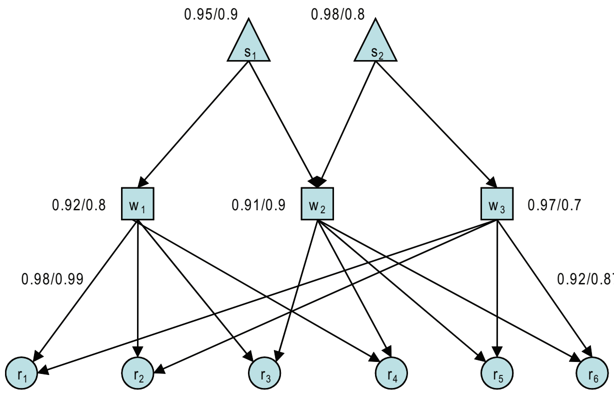

Fig. 2. A two-echelon stochastic supply chain $\mathcal{S}_e$

needs to be decided by checking all the possible resolutions of nondeterminism. PRISM supports two variants of P for model-checking Markov Decision Process: P_{max} represents the best-case scenario in which a resolution of nondeterminism maximizes the probability that a path formula holds, and P_{min} represents the worst-case scenario. Note that in our stochastic supply chain modeling, routing decisions are modeled as nondeterministic transitions. We use P_{max} to force PRISM to search for the best routing strategy that improves the stochastic performance of a supply chain.

III. Model Stochastic Supply Chains

The first step of model checking is to model a subject in a formalism that facilitates automated analysis. Before we introduce the formalism for modeling stochastic behaviors of a supply chain, we discuss the intuition behind the formal modeling. To model the dynamics of a supply chain, we need to identify its states and transitions. A typical supply chain consists of suppliers, warehouses, retailers, and routes connecting them. Figure 2 shows an example of a two-echelon supply chain network. The label of an component indicates the probabilities of an element failing and recovering. For example, the label for the warehouse w_1 indicates that at any time it may fail with the probability of 0.92, and when that happens, it may go back to the operational mode with the probability of 0.8 at any time afterwards. By default, an element without a label is always operational.

The purpose of a supply chain is to transfer goods. The dynamics of a supply chain is characterized by merchandise flowing among it. To identify the state of a supply chain, we need to consider the movement of merchandise and the status of each individual element of the supply chain. For instance, suppose that the supply chain in Figure 2 carries two products, G_A and G_B. The state of the supply chain can be decided by A's and B's locations, and the status of each element, i.e., if an element is still operational. The stochastic model we are about to propose is the synchronized composition of element models and merchandise flow models, each of which is represented by an Extended Markov Decision Process (EMDP).

Definition 2 (Element EMDP): An element EMDP (E-EMDP) is an EMDP $\langle\langle w\rangle, w_0, P\rangle$, where,

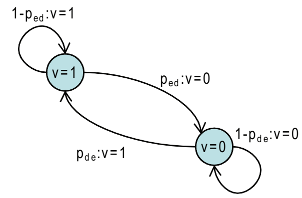

Fig. 3. The state space and transitions of the E-EMDP for the warehouse w_1

1) w is the only variable and its domain is Boolean. Semantically, w stands for whether the element is operational.
2) The probabilistic transition function P is defined as follows: $P(\mathbf{T}) = \{\langle \mathbf{F}, p_{ed}\rangle, \langle \mathbf{T}, (1-p_{ed})\rangle\}$ and $P(\mathbf{F}) = \{\langle \mathbf{T}, p_{de}\rangle, \langle \mathbf{F}, (1-p_{de})\rangle\}$.
3) w_0 is the initial value of w.

We call w the status variable, p_{ed} operational probability, and p_{de} recovering probability. □

Intuitively, an element EMDP represents a two-state Markov Decision Process. An element can either be operational or not. If it is operational, it may fail with the probability of p_{ed} next time. If it is already non-operational, next time it may recover with the probability of p_{de}. Figure 3 shows the state space and transitions implied by the element EMDP for the warehouse w_1 in 2.

Definition 3 (Merchandise EMDP): Let $\mathcal{S}$ be a stochastic supply chain with k elements, and let w_i be the status variable of the i-th element. A merchandise EMDP (M-EMDP) for a product carried by $\mathcal{S}$ is a EMDP $\langle\langle v\rangle, \langle w_1 \cdots w_k\rangle, f_0, N\rangle$, where,

1) v is the only variable and its domain is the set of facilities in $\mathcal{S}$. A facility can be one of the following elements: a supplier, a warehouse, or a retailer.
2) The nondeterministic transition function N is defined as follows: for every facility f and every route ff' emanating from it, $N(\langle f\rangle, \langle w_1 \cdots w_k\rangle)$ is,
   - $\{\langle f\rangle, \langle f'\rangle\}$, if $w_{e_f} = w_{e'_f} = w_{e_{ff'}} = \mathbf{T}$, or;
   - $\{\langle f\rangle\}$, otherwise.
3) f_0 is the initial value of v.

We refer to v as the location variable. □

A merchandise EMDP represents how a product is transported through a stochastic supply chain. The location of the product is denoted by its location variable v. v's initial value f_0 represents the manufacturing facility for the product. The nondeterministic transition function specifies how the next location is chosen: the product can either stay at its current location f, or in case that f, f', and the route ff' are all operational, the product may also be transferred to f'. Unlike the probabilistic transition function in an E-EMDP, the transition function of an M-EMDP is nondeterministic: the model does not specify in what probability the next location is chosen from a list of eligible locations. A probabilistic model checker has to consider all the possible ways of resolving nondeterminism. In Section IV we take advantage of such capability and ask a probabilistic model checker to search for

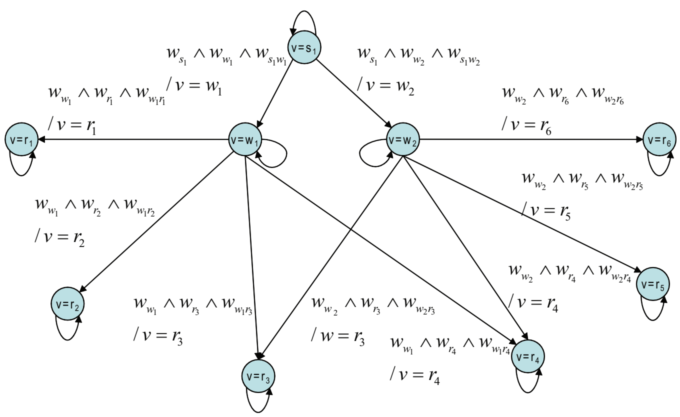

Fig. 4. The state space and transitions of the M-EMDPmodel for a product G_A manufactured in s_1.

the best scenario for resolving nondeterminism. The answer produced by the model checker implies the optimal strategy for scheduling merchandise flow.

Figure 4 shows the the state space and transitions implied by the M-EMDP for a product manufactured at the facility s_1. Note that it has a similar structure as the subset of its underlying supply chain in 2. This is because M-EMDP depicts the flow of a product in a supply chain and it shall have a similar structure as part of the supply chain the product may go through.

Definition 4 (Stochastic Merchandise Flow Model): Let $\mathcal{S}$ be a stochastic supply chain and $\mathcal{P}$ a set of products transported in $\mathcal{S}$, the stochastic merchandise flow (SMF) model of $\mathcal{S}$ is a synchronized parallel composition of EMDPs in $\mathcal{E} \cup \mathcal{M}$, where $\mathcal{E}$ is the set of all the E-EMDPs for elements of $\mathcal{S}$ and $\mathcal{M}$ is the set of all the M-EMDPs for products in $\mathcal{P}$. □

We use Stochastic Merchandise Flow Model in Definition 4 to model the dynamics of a stochastic supply chain. A SMF model is a synchronized parallel composition of all the E-EMDPs and M-EMDPs. We write $N_{\mathcal{S}} = ||_{n \in \mathcal{E} \cup \mathcal{M}} \, n$ for the SMF model of a stochastic supply chain $\mathcal{S}$, where $\mathcal{E}$ and $\mathcal{M}$ are the set of E-EMDPs and the set of M-EMDPs in $\mathcal{S}$, respectively. States of a SMF model are identified by the evaluation of status variables in E and location variables in M. Transitions are synchronized compositions of transitions of all the E-EMDPs in E and M-EMDPs in M. That is, $\langle w_1, \cdots, w_i, v_1, \cdots, v_j \rangle \rightarrow \langle w'_1, \cdots, w'_i, v'_1, \cdots, v'_k \rangle$ is a transition of $N_{\mathcal{S}}$ if and only if for every l such that $1 \leq l \leq i$, $w_l \rightarrow w'_l$ is a transition of $e_l \in \mathcal{E}$ and for every k such that $1 \leq k \leq j$, $v_k \rightarrow v'_k$ is a transition of $m_k \in \mathcal{M}$. Intuitively, a transition of $N_{\mathcal{S}}$ represents a discrete step in supply chain operations. During the discrete step, an E-EMDP may flip its status variable with a given probability, and an M-EMDP may nondeterministically decide what will be the next location of the product it represents.

IV. Analyze stochastic supply chains

We use probabilistic model checking to automate the analysis of stochastic supply chain models. As we discussed in Section III, the underlying formalism for stochastic supply chain models is EMDP, an extension of Markov Decision Process. Traditionally decision procedures for Markov Decision Process use dynamic programming technique and they are usually customized for targeted problem domains. Our choice of using probabilistic model checking as underlying analysis technology brings us several benefits: first, recent advances in probabilistic model checking provide efficient symbolic decision procedures. Symbolic probabilistic model checkers such as PRISM [4] use sophisticated symbolic techniques including Binary Decision Diagrams (BDD) and Multi-Terminal Binary Decision Diagrams (MTBDD) [8]. By using probabilistic model checking, we are able to leverage benefits of efficient decision procedures developed for verifying large-scale computer-based systems. Second, traditionally decision procedures analyze Markov Decision Processes by attaching to each transitions or states a *reward* and then optimizing a reward-based cost function. Although the reward-based approach is appealing, it also has its limitations. In comparison, a probabilistic model checker provides a generic decision procedure which not only supports the reward-based analysis approach but also enables us to specify sophisticated stochastic and temporal properties in a temporal logic. For example, the probabilistic model checker PRISM uses the probabilistic Computational Tree Logic (PCTL), and we can specify a probabilistic temporary property P_{reach}: what is the possibility that *a product A can be delivered to a retailer r_i within 4 days after arriving a warehouse w_k*.

As part of analysis activities, we need to specify stochastic properties in a temporal logic, in our case, in PCTL. The basic building blocks of a PCTL formula are atomic propositions. Semantically an atomic proposition refers to a set of states in which by definition the proposition holds. In practices, we use an atomic proposition to label a set of states of special interest in analysis. Since a state is presented by a valuation of variables in EMDP, we may define an atomic proposition using a predicate over variables. For instance, to specify states of a product A "*arriving a warehouse* w_k", we define an atomic proposition $\mathcal{A}_{w_k} = (v_A = w_k)$, where v_A is A's location variable. Note that the predicate $v_A = w_k$ only constrains v_A's value, therefore, $\mathcal{A}_{w_k}$ specifies a set of states. That is, $\mathcal{A}_{w_k}$ holds on any state in which A is in the warehouse w_k. Other state information, whether w_k is operational, is irrelevant to the semantics of $\mathcal{A}_{w_k}$.

The probabilistic model checker PRISM supports an extension of PCTL using bounded path formulae, which allows us to specify the number of discrete steps a property shall hold. As part of modeling and analysis tasks, one needs to decide the semantics of a discrete step in a SMF model based on the planning horizon of supply chain operations. For instance, if operations such as shipment are scheduled in term of days, a discrete step can be one day.

PCTL supports a set of temporal operators in the Computation Tree Logic (CTL) such as U (until), R (release), X (next), and their bounded version. Using PCTL, one may also encode the property such as in what probability a property holds on a

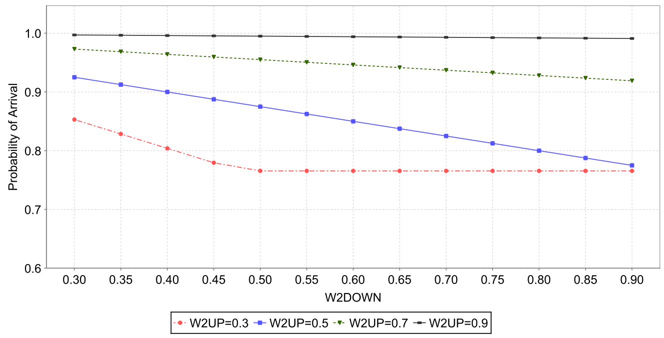

Fig. 5. The stochastic performance of the supply chain $\mathcal{S}_c$

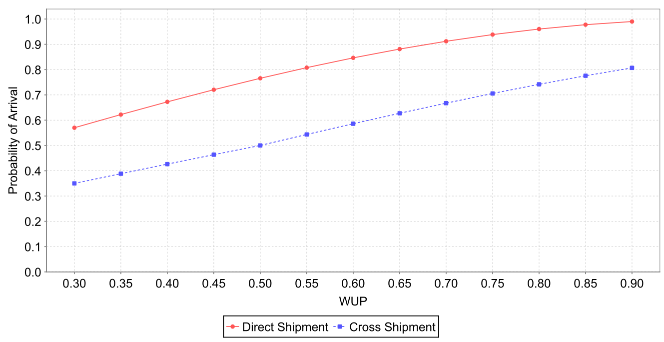

Fig. 6. The stochastic performance of $\mathcal{S}_{cw}$, a cross-warehouse shipment variant of $\mathcal{S}_c$ w.r.t. the direct shipment ($\mathsf{P}_{max=?}[G((v_1 = w_3 \wedge v_2 = w_1) \to F^{\leq 4}\mathcal{A}_{arrived})]$)

system. For example, $\mathsf{P}_{max\geq T}(\mathbf{G}(v_A = w_k)F^{\leq 3}(v_A = r_i))$ tests the threshold of probability that the product A can be delivered to r_i 3 days after it arrives at the warehouse w_k. In addition, PRISM supports the use of "?" in place of a real number to query in what probability a property holds. In PRISM, the PCTL formula for the property P_{reach} is $P_{reach} = \mathsf{P}_{max\geq ?}(\mathbf{G}(v_A = w_k)F^{\leq 3}(v_A = r_i))$. After checking P_{reach} on a stochastic supply chain model, PRISM returns the probability that the property holds for the model.

V. Experiments

To assess the feasibility of our model-checking-based analysis technique and evaluate its capability, we use it to check a variety of stochastic supply chain models. In this section we report our experiments on supply chain models with three different topologies. All the experiments are carried out on a Windows Vista machine with a 2.0 GHz Intel Core 2 Duo T7300 processor and 1 GB memory. We use PRISM version 3.2 beta in our experiments.

Figure 1 measures the stochastic performance of the supply chain $\mathcal{S}_c$ in Figure 2. In the experiment, we use two products A and B, produced by the suppliers s_1 and s_2 respectively, to measure $\mathcal{S}_c$'s performance under different configurations. A and B are destined for the retailer r_3 and r_5, respectively. We measure their probabilities of arrivals under different configurations: W2DOWN and W2UP stand for the failure and recovery probabilities of the warehouse w_2, and these probabilities vary in our experiment. Other warehouses, w_1 and w_3, have the fixed failure and recovery probabilities of 0.5. To express the property that the products arrive at their destination within 4 steps, we use the PCTL formula $\mathsf{P}_{max=?}[F^{\leq 4}(v_A = r_3) \wedge (v_B = r_5)]$, where v_A and v_B are A's and B's location variables with the initial values of s_1 and s_2, respectively. Figure 5 shows that a robust warehouse, indicated by a low failure probability and a high recovery probability, increases the probability that a product arrives at its destination on time. Model construction and model checking took 0.111 second and 0.059 second for this experiment in this experiment..

Figure 2 measures the stochastic performance of a cross-warehouse shipment variant of $\mathcal{S}_c$. The variant, $\mathcal{S}_{cw}$ adds routes among warehouses and enables traffic among neighboring warehouses. $\mathcal{S}_{cw}$ removes the routes s_1w_2 and s_2w_3 from $\mathcal{S}_c$. We test two different settings for products A and B. In the *direct shipment* A and B are manufactured by s_1 and s_2, and shipped to r_3 and r_5. The PCTL property for the direct shipment setting is $\mathsf{P}_{max=?}[F^{\leq 4}(v_A = r_3) \wedge (v_B = r_5)]$. In the *cross-warehouse shipment* A and B are shipped to r_5 and r_3. The PCTL property for the cross-warehouse shipment setting is $\mathsf{P}_{max=?}[F^{\leq 4}(v_A = r_5) \wedge (v_B = r_3)]$. Note that all the routes from s_1 to r_5 or from s_2 to r_3 have to involve some cross-warehouse traffic. WUP is the failure probability of the warehouses. Their recovery probability is fixed at 0.5. Figure 6 shows that a quick recovery, indicated by an arising recovery probability, improves the probability that the products arrive at their destinations on time. It also indicates that the cross-warehouse shipment is affected most by warehouses' recovery probability since it involves more stops at warehouses than the direct shipment does. Model construction and model checking took 0.164 second and 0.257 second in this experiment.

Figure 3 measures the stochastic performance of a three-echelon variant of $\mathcal{S}_c$. The variant, $\mathcal{S}_{e3}$ adds one more layer warehouses underneath the existing layer of warehouses. A warehouse can ship to its child and its immediate successor on the second level of warehouses. This extra layer gives $\mathcal{S}_{e3}$ the capability of re-routing some shipment if necessary and it also increases complexity in supply-chain design. Like in the first experiment, we measure the probabilities of arrivals under different configurations but with some variances: instead

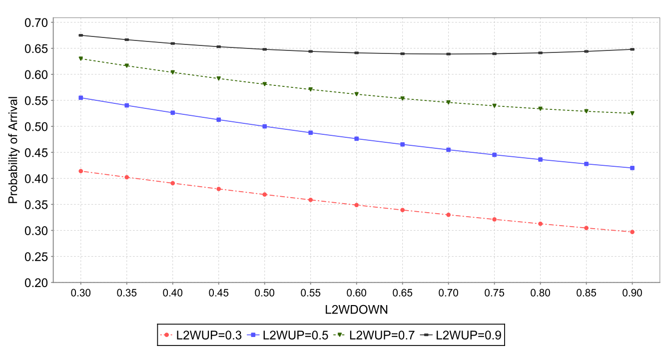

Fig. 7. The stochastic performance of three echelon variant of $\mathcal{S}_c$ w.r.t. the PCTL property $\mathsf{P}_{max=?}[G((v_1 = w_2 \wedge v_2 = w_2) \to F^{\leq 4}\mathcal{A}_{arrived})]$

of changing the probabilities for the warehouse w_2 in Figure 5, we change the probabilities for the entire second layer of warehouses. We also require that both products stop by the warehouse w_2 so we can take into account $\mathcal{S}_{e3}$'s ability to re-reroute products to their destination. The property is encoded in PCTL as $\mathsf{P}_{max=?}[G((v_A = w2) \wedge (v_B = w2) \rightarrow F^{\leq 4}(v_A = r_3) \wedge (v_B = r_5))]$. Figure 7 shows that an improvement on the robustness of warehouses will contribute positively to the probability of arrival. Model construction and model checking took 0.13 second and 1.497 second in this experiment.

VI. Conclusion

We proposed a novel computational framework for modeling and analyzing stochastic supply chains. Our introduction of probabilistic model checking to the domain of supply chain risk management improves the efficiency and scalability of existing decision procedures. We introduced a formal modeling framework SMF for stochastic supply chains. Based on an extension of Markov Decision Process, SMF rigorously defines the syntax and semantics of a supply chain and facilitates formal analysis. We also discussed the procedure we proposed for using a probabilistic model checker to analyze a stochastic supply chain model. Particularly we discuss how one can express in the temporal logic PCTL the stochastic properties of a supply chain. Our experiments demonstrate the capability of our approach on stochastic supply chains of different configurations.

This work may be extended in several directions. For example, Our focus in this research is on transportation processes in supply chain operations. In the future we want to extend our approach to include other elements of a supply chain such as manufacturing process. SMF we introduced is also capable of specifying interactions among different elements in supply chain. In the future we want to use SMF to study such interactions and their implication on risks in a supply chain operation.

References

[1] A. Bianco and L. de Alfaro. Model checking of probabilistic and nondeterministic systems. In P. Thiagarajan, editor, *Proc. 15th Conference on Foundations of Software Technology and Theoretical Computer Science*, volume 1026 of *LNCS*, pages 499–513. Springer, 1995.

[2] F. Chen, A. Federgruen, and Y. Zheng. Coordination mechanisms for a distribution system with one supplier and multiple retailers. *Management Science*, 47(5):693–708, 2001.

[3] John Heath, Marta Kwiatkowska, Gethin Norman, David Parker, and Oksana Tymchyshyn. Probabilistic model checking of complex biological pathways. *Theor. Comput. Sci.*, 391(3):239–257, 2008.

[4] A. Hinton, M. Kwiatkowska, G. Norman, and D. Parker. Prism: A tool for automatic verification of probabilistic systems. In H. Hermanns and J. Palsberg, editors, *Proc. 12th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'06)*, volume 3920 of *LNCS*, pages 441–444. Springer, March 2006.

[5] M. Kwiatkowska. Quantitative verification: Models, techniques and tools. In *Proc. 6th joint meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE)*, pages 449–458. ACM Press, September 2007.

[6] Hau Lee and Seungjin Whang. Decentralized multi-echelon supply chains: Incentives and information. *Manage. Sci.*, 45(5):633–642, 1999.

[7] M. L. Puterman. *Markov decision processes, discrete stochastic dynamic programming*. Wiley-Interscience, 1994.

[8] F. Wang and M. Kwiatkowska. An MTBDD-based implementation of forward reachability for probabilistic timed automata. In D. Peled and Y.-K. Tsay, editors, *Proc. 3rd International Symposium on Automated Technology for Verification and Analysis (ATVA'05)*, volume 3707 of *LNCS*, pages 385–399. Springer, 2005.