Discovering Stochastic Causal Nets

Tian Li
The University of Melbourne, Australia
RWTH Aachen University, Germany
tian.li.2@unimelb.edu.au

Sander J.J. Leemans
RWTH Aachen University, Germany
Fraunhofer, Germany
s.leemans@bpm.rwth-aachen.de

Artem Polyvyanyy
The University of Melbourne
Parkville, VIC 3010, Australia
artem.polyvyanyy@unimelb.edu.au

***Abstract*—Process mining leverages event logs extracted from information systems to generate insights into the business processes of organizations. These insights are enhanced by explicitly accounting for the frequency of behavior captured in stochastic process models constructed from event logs. Causal nets are an elegant declarative process modeling formalism that relies on a small number of modeling constructs, yet is expressive. In this paper, we extend this formalism to the stochastic setting, that is, to allow the extended nets to capture the likelihoods of the observed process. We also propose a stochastic causal net discovery approach using Markovian abstraction. Our approach begins with a standard causal net model generated by a control flow discovery algorithm, and then employs optimization techniques to determine optimal binding weights. These weights enable the stochastic interpretation of the model to closely approximate the Markovian abstraction of the original event log. Our technique has been implemented and made publicly available. The evaluation based on this implementation demonstrates the feasibility of the technique. Compared to baseline models, the discovered models achieve noticeable improvements in the quality of stochastic conformance.**

***Index Terms*—Process mining, Stochastic process discovery, Stochastic causal nets**

I. Introduction

Stochastic process mining specifically focuses on techniques that incorporate the frequency and probability of different process behavior. Rather than simply capturing the control-flow structure, stochastic process discovery constructs process models that explicitly represent the stochastic nature of the observed processes. This perspective is crucial for organizations, because it enables them to differentiate between routine operations and exceptional cases. Without this capability, business analysts risk misallocating resources by directing attention to rare behaviors that have a minimal impact on overall process performance [10].

Among various modeling formalisms, causal nets (C-nets) are employed by multiple process discovery techniques, such as Flexible Heuristic Miner [18] and Fodina Miner [5]. Unlike other conventional process models, C-nets directly capture the routing logic through input and output bindings for each activity, without using model elements such as silent activities, places, or gateways. However, despite their effectiveness in control-flow modeling, current C-net discovery techniques do not reflect the stochastic nature of process behavior. This limitation undermines their potential for business process improvement, as the models cannot reproduce the probabilistic characteristics of the observed processes.

To address this challenge, stochastic process discovery methods construct process models that associate each trace with probability, to reflect their expected occurrence in future process executions. Existing stochastic discovery techniques first apply a standard algorithm to obtain a control flow model and then convert it into a stochastic model based on the trace probabilities of log [7–9]. Although empirical studies have demonstrated their effectiveness when applied to real-world event logs, they often disregard partial mismatches between traces from event log and process model during discovery.

In this paper, we introduce a novel extension by incorporating a stochastic perspective into the modeling with C-nets. First, we provide formal semantics for stochastic C-nets, where probability distributions over traces are determined by the weights of input and output bindings. Second, given an event log and a C-net, we apply a Markovian abstraction to account for local subtrace similarities, which complements existing techniques by addressing their limitation of treating zero-probability log traces as entirely incompatible with models. Then, stochastic discovery is formulated as an optimization problem, which constructs a stochastic C-net that maximizes the stochastic conformance measured through the Markovian abstractions of log and model. Third, we demonstrate the feasibility of the technique with several real-life event logs.

The paper is organized as follows: and Section II states preliminaries. In Section III, we introduce stochastic C-nets, and subsequently present two stochastic discovery algorithms in Section IV. We evaluate the approach in Section V, and Section VI discusses related work. Finally, Section VII concludes the paper.

II. Preliminaries

This section presents several concepts and definitions to understand the main methods of the paper. Given a set X, $\mathcal{P}(X)$ denotes the power set of X. Given two sets X_1 and X_2, $X_1 \backslash X_2 = \{x \mid x \in X_1 \land x \notin X_2\}$ is the set of elements in X_1 but not in X_2. A multiset is a collection that allows multiple occurrences of its elements. We specify a multiset M over X as a function $M : X \rightarrow \mathbb{N}$ that maps the elements in X to natural numbers (including zero). For example, $M = [b^4, c^5, d]$ is a multiset with ten elements: four b's, five c's, and one d. The set of all multisets over X is denoted by $\mathbb{B}(X)$. The union of two multisets M_1 and M_2 is denoted by $M_1 \uplus M_2$. By $M_1 \subseteqq M_2$, we denote the fact that $\forall_{x \in X} M_1(x) \leq M_2(x)$.

If $M_1 \Subset M_2$, then $M_3 = M_2 \setminus M_1$ is the multiset difference, such that $\forall_{x \in X} M_3(x) = M_2(x) - M_1(x)$.

An *event log* describes the observed behavior of system, which is a multiset of traces, where a *trace* is a finite sequence of activities. For instance, $L = [\langle b, c, e\rangle^{10}, \langle b, d, e\rangle^{10}, \langle b, c, d, e\rangle^{30}, \langle b, d, c, e\rangle^{50}]$ is an event log with 100 traces. We use the $\otimes$ operator to concatenate an element to the end of a sequence, e.g. $\langle b, c, d\rangle \otimes e = \langle b, c, d, e\rangle$. For a vector $\vec{x}$, x_i denotes the element at the i-th position in the vector.

A. Causal Nets

Process models constructed by heuristic dependency-based process discovery techniques are often expressed as causal nets (C-nets) [5]. A C-net is a directed graph in which nodes represent activities and edges represent causal relations. Each activity has a set of input bindings and a set of output bindings.

Definition 1 (Causal Nets [1])**.** Let A be a finite set of activities, and $AS = \{Y \subseteq \mathcal{P}(A) \mid Y = \{\emptyset\} \vee \emptyset \notin Y\}$. A causal net is a tuple $N = (A, D, I, O, a_i, a_f)$, where $D \subseteq A \times A$ is the dependency relation, $I \in A \to AS$ defines the set of possible input bindings per activity, $O \in A \to AS$ defines the set of possible output bindings per activity, $\{a_i\} = \{a \in A \mid I(a) = \{\emptyset\}\}$ is the set of initial activities, and $\{a_f\} = \{a \in A \mid O(a) = \{\emptyset\}\}$ is the set of final activities.

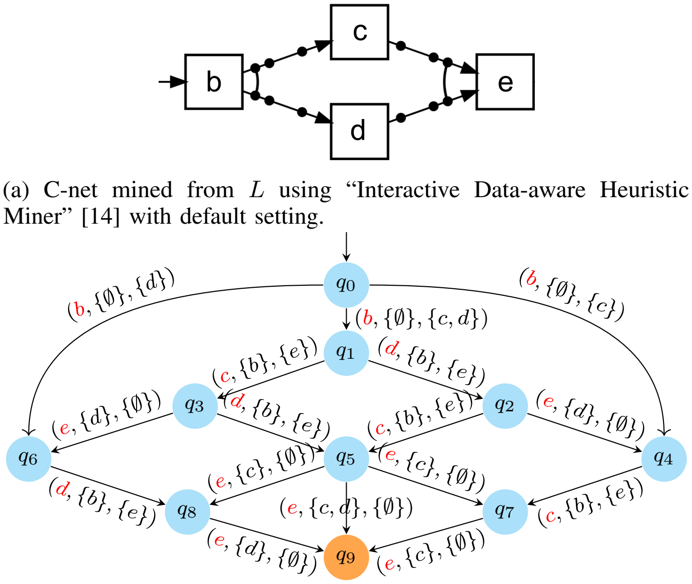
In a C-net, input and output bindings allow for split and join behavior similar to those defined in other modeling formalisms. Fig. 1a presents a C-net mined from L, where $A = \{b, c, d, e\}$ is the set of activities, $D = \{(b, c), (b, d), (c, e), (d, e)\}$ is the dependency relation, $I(b) = \{\emptyset\}$, $O(b) = \{\{c\}, \{d\}, \{c, d\}\}$, $I(c) = I(d) = \{\{b\}\}$, $O(c) = O(d) = \{\{e\}\}$, $I(e) = \{\{c, d\}, \{c\}, \{d\}\}$, and $O(e) = \{\emptyset\}$. Activity b is the initial activity, and e is the final activity.

Definition 2 (Activity Bindings)**.** Let $C = (A, D, I, O, a_i, a_f)$ be a C-net. $B = \{(a, as_I, as_O) \in A \times \mathcal{P}(A) \times \mathcal{P}(A) | as_I \in I(a) \wedge as_O \in O(a)\}$ is the set of *activity bindings*.

An activity binding is a triple (a, as_I, as_O) that indicates the occurrence of an activity a with an input binding as_I and an output binding as_O. A *binding sequence* $\gamma \in B^*$ is a sequence of activity bindings, which can be projected to a trace using a projection function $\omega \colon B^* \to A^*$. For the example C-net, $\gamma_1 = \langle(b, \emptyset, \{c, d\}), (c, \{b\}, \{e\}), (d, \{b\}, \{e\}), (e, \{c, d\}, \{\emptyset\})\rangle$ is a possible binding sequence that can be projected to trace $\langle b, c, d, e\rangle$.

Definition 3 (States)**.** Let $N = (A, D, I, O, a_i, a_f)$ be a C-net. $S = \mathbb{B}(A \times A)$ is the state space of N. A state $s \in S$ is a multi-set of *pending obligations*. Function $\psi \in B^* \to S$ is defined inductively: $\psi(\langle\rangle) = [\,]$ and $\psi(\sigma \oplus (a, as_I, as_O)) = (\psi(\sigma) \setminus (as_I \times \{a\})) \uplus (\{a\} \times as_O)$ for any binding sequence $\sigma \oplus (a, as_I, as_O) \in B^*$. $\psi(\sigma)$ is the state after executing the binding sequence σ.

A pending obligation of a state is a pair. Initially, there are no pending obligations, as no output bindings have been *enacted* without having corresponding input bindings. By B_s, we denote the set of all activity bindings *enacted* in state s, that is, $\exists_{\sigma \in B^* \text{ s.t. } s = \psi(\sigma)}\ B_s = \{(a, as_I, as_O) \mid as_I \times \{a\} \Subset \psi(\sigma)\}$. *Executing* an *enacted* activity binding removes pending obligations, and introduces new pending obligations. For instance, the initial state of the C-net shown in Fig. 1a is $s_0 = [(\emptyset, b)]$. If activity binding $(b, \emptyset, \{c\})$ is executed, we have $\psi(\langle(b, \emptyset, \{c\})\rangle) = \psi(\langle\rangle) \setminus (\emptyset \times \{b\}) \uplus (\{b\} \times \{c\}) = [\,] \setminus [\,] \uplus [(b, c)] = [(b, c)]$. Given B_s for state s, the set of enacted input bindings is denoted as $I_s = \{as_I \mid \exists_{(a, as_I, as_O) \in B_s}\}$.


b
c
d
e
q_0
q_1
q_2
q_3
q_4
q_5
q_6
q_7
q_8
q_9
$(b, \{\emptyset\}, \{d\})$
$(b, \{\emptyset\}, \{c\})$
$(b, \{\emptyset\}, \{c, d\})$
$(c, \{b\}, \{e\})$
$(d, \{b\}, \{e\})$
$(e, \{d\}, \{\emptyset\})$
$(d, \{b\}, \{e\})$
$(c, \{b\}, \{e\})$
$(e, \{d\}, \{\emptyset\})$
$(e, \{c\}, \{\emptyset\})$
$(e, \{c\}, \{\emptyset\})$
$(d, \{b\}, \{e\})$
$(e, \{c, d\}, \{\emptyset\})$
$(c, \{b\}, \{e\})$
$(e, \{d\}, \{\emptyset\})$
$(e, \{c\}, \{\emptyset\})$

(a) C-net mined from L using "Interactive Data-aware Heuristic Miner" [14] with default setting.

(b) In the reachability graph, $q_0 = [(\emptyset, b)]$, $q_1 = [(b, c), (b, d)]$, $q_2 = [(b, c), (d, e)]$, $q_3 = [(b, d), (c, e)]$, $q_4 = [(b, c)]$, $q_5 = [(c, e), (d, e)]$, $q_6 = [(b, d)]$, $q_7 = [(c, e)]$, $q_8 = [(d, e)]$, and $q_9 = [\ ]$.

Fig. 1: A C-net and its reachability graph.

A *valid* binding sequence models an execution path starting from the initial state and ending with the removal of all obligations created during execution. Consider γ_1, we have:

$$\begin{aligned}
&\psi(\langle\rangle) = [],\\
&\psi(\langle(b, \emptyset, \{c, d\})\rangle) = [(b, c), (b, d)],\\
&\psi(\langle(b, \emptyset, \{c\}), (c, \{b\}, \{e\})\rangle) = [(c, e), (b, d)],\\
&\psi(\langle(b, \emptyset, \{c\}), (c, \{b\}, \{e\}), (d, \{b\}, \{e\})\rangle) = [(c, e), (d, e)], \text{ and}\\
&\psi(\langle(b, \emptyset, \{c\}), (c, \{b\}, \{e\}), (d, \{b\}, \{e\}), (e, \{c, d\}, \{\emptyset\})\rangle) = [].
\end{aligned}$$

Moreover, a process model is sound if it is free of deadlocks, livelocks, and other obvious anomalies [2]. In this paper, we only consider sound C-nets, and the behavior of C-nets is limited to valid binding sequences. The execution semantics of a C-net are described by its reachability graph.

Definition 4 (Reachability Graphs)**.** The reachability graph (RG) of a C-net is a tuple (Q, A, δ, q_0, F) where Q is a finite set of states, A is a finite set of actions, $\delta : Q \times A \to Q$ is the transition function, q_0 is the initial state, and $F \subseteq Q$ is the set of final states.

The reachability graph of a C-net has the following features: (a) labels are built from activity bindings; (b) states correspond to states of the C-net that are reachable from the initial state; (c) the initial state corresponds to the initial state of the C-net; (d) the final state has no pending obligation. For instance, the RG of the example C-net is illustrated in Fig. 1b.

B. Stochastic Languages and Stochastic Conformance

Definition 5 (Multiset of k-Trimmed Subtraces)**.** Let A be a finite set of activities, and $\sigma \in A^*$ be a trace. The multiset of k-trimmed subtraces for σ is defined recursively as:

$$M^k_\sigma = \begin{cases} \{\sigma\} & \text{if } |\sigma| < k \\ \{\sigma^{1\to k}\} \uplus M^k_{\sigma^{2\to|\sigma|}} & \text{otherwise.} \end{cases}$$

For instance, given a trace $\sigma = \langle b, c, c, c\rangle$, it holds that $M^2_\sigma = [\langle b, c\rangle,\ \langle c, c\rangle^2]$, $M^3_\sigma = [\langle b, c, c\rangle,\ \langle c, c, c\rangle]$, and $M^k_\sigma = [\langle b, c, c, c\rangle]$ for all $k \geq 4$.

Definition 6 (Stochastic Languages)**.** Let A be a finite set of *activities* and let A^* be the set of all finite sequences of activities (*traces*) over A. Then, a stochastic language l is a function that maps each trace in A^* to a probability, that is, $l : A^* \to [0, 1]$ such that $\sum_{\sigma\in A^*} l(\sigma) = 1$.

A stochastic language is an assignment of probabilities to traces so that the total probabilities sum up to one. It can be used to encode the relative probability of observing a trace in an event log or a stochastic process model. Consider event log L, $l_L(\langle b, c, e\rangle) = 0.1$, $l_L(\langle b, d, e\rangle) = 0.1$, $l_L(\langle b, c, d, e\rangle) = 0.3$, and $l_L(\langle b, d, c, e\rangle) = 0.5$.

A stochastic conformance measure compares the stochastic languages of an event log and a stochastic process model. In this paper, we use unit Earth Movers' stochastic conformance for stochastic process discovery.

Definition 7 (Unit Earth Movers' Stochastic Conformance [11])**.** Let L be an event log, and N be a stochastic process model. Their unit Earth Movers' Stochastic Conformance (uEMSC) is defined as:

$$\text{uEMSC}(L, N) = 1 - \sum_{\sigma\in A^*} \max(L(\sigma) - N(\sigma), 0).$$

uEMSC ranges from 0 to 1, where 1 indicates perfect conformance and 0 indicates the worst conformance. Although it can be computed efficiently in practice, partial trace mismatches are not considered. Consider two traces of 10 events, even if they differ only in their end events, uEMSC considers them to be completely different. However, these two traces can be classified as almost equivalent and measured with a lower deviation value. Markovian-based abstraction addresses this partial matching issue by accounting for subtraces within a stochastic language.

Definition 8 (K-th-order Stochastic Markovian Abstractions [16])**.** Let l be a stochastic language and $k \geq 2$. The k-th-order stochastic Markovian abstraction of l is a stochastic language $m^k_l : A^* \to [0, 1]$ defined as:

$$m^k_l(\gamma) = \frac{f^k_l(\gamma)}{\sum_{\gamma'\in A^*} f^k_l(\gamma')}, \text{ such that:} \tag{1}$$

$$f^k_l(\gamma) = \sum_{\sigma\in A^*} l(\sigma) \cdot M^k_\sigma(\gamma).$$

In Eq. (1), f^k_l represents the expected number of occurrences of l's k-trimmed subtraces. The k-th order stochastic Markovian abstraction m^k_l yields a probability distribution over subtraces, which is the normalization of f^k_l with the sum of the occurrences of all k-trimmed subtraces.

For instance, subtrace $\langle b, c\rangle$ has a presence in traces $\langle b, c, e\rangle$ and $\langle b, c, d, e\rangle$ in log L, thus $M^2_{\langle b,c,e\rangle}(\langle b, c\rangle) = M^2_{\langle b,c,d,e\rangle}(\langle b, c\rangle) = 1$, and the relative frequency of $\langle b, c\rangle$ is $f^2_L(\langle b, c\rangle) = 0.1 * 1 + 0.3 * 1 = 0.4$. Consider other subtraces of length 2 from L, we have $f^2_L(\langle b, d\rangle) = 0.1 * 1 + 0.5 * 1 = 0.6$, $f^2_L(\langle c, d\rangle) = 0.3 * 1 = 0.3$, $f^k_L(\langle d, c\rangle) = 0.5 * 1 = 0.5$, $f^2_L(\langle c, e\rangle) = 0.1 * 1 + 0.5 * 1 = 0.6$, $f^2_L(\langle d, e\rangle) = 0.1 * 1 + 0.3 * 1 = 0.4$. For normalization, we get $f_{sum} = \sum_{\gamma'\in A^*} f^2_L(\gamma') = 0.4 + 0.6 + 0.3 + 0.5 + 0.6 + 0.4 = 2.8$, and $m^2_L(\gamma)$ for each γ is derived by dividing each $f^2_L(\gamma)$ with f_{sum}, e.g. $m^2_L(\langle b, c\rangle) = {}^{0.4}/_{2.8} = {}^{1}/_{7}$. The 2-th-order stochastic Markovian abstraction for L is $[\langle b, c\rangle^{1/7}, \langle b, d\rangle^{3/14}, \langle c, d\rangle^{3/28}, \langle c, e\rangle^{3/14}, \langle d, c\rangle^{5/28}, \langle d, e\rangle^{1/7}]$.

Definition 9 (Markovian-based Unit Earth Movers' Conformance)**.** Let l and n be two stochastic languages, their k-th order Markovian-based unit Earth Mover's Stochastic Conformance (uEMSC^k) is:

$$\text{uEMSC}^k(l, n) = 1 - \sum_{\sigma\in A^*} \max(m^k_l(\sigma) - m^k_n(\sigma), 0).$$

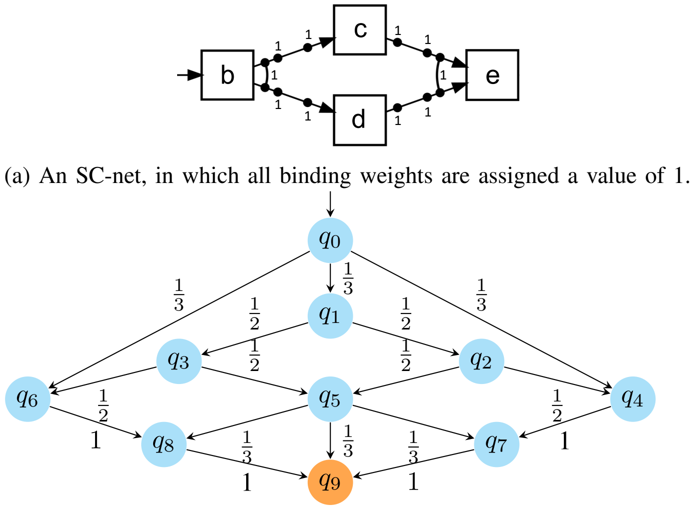
Given two stochastic languages, uEMSC^k compares their probabilities of subtraces of length k. If k approaches $+\infty$, uEMSC^k approaches uEMSC. In contrast, a smaller k allows uEMSC^k to account for shorter subtraces.

Finally, we introduce the stochastic deterministic finite automaton to define the semantics of the stochastic C-net used in the subsequent section.

Definition 10 (Stochastic Deterministic Finite Automaton)**.** A stochastic deterministic finite automaton (SDFA) is a tuple $(Q, A, E, \delta,\ q_0,\ \pi)$ where Q is a set of states, A is a finite set of actions, $E : Q \times A \to Q$ is the transition function, $\delta : Q \times A \to [0, 1]$ is the probability function for transition, q_0 is the initial state and $\pi : Q \to [0, 1]$ is the function defining the final probability of each state, where $\forall q \in Q : \sum_{(l,q')\in(A\times Q)} \delta(q, l) + \pi(q) = 1$.

An SDFA can be used to define the reachability graph of the stochastic extended C-nets. By $\delta(q \mid \gamma)$, we denote the probability of a sequence of activities γ that starts in state $q \in Q$. We denote the transition matrix of SDFA as $\hat{\Delta}$, the probability of visiting every state as column vector $\vec{x}$, and let $\hat{I}$ be the identity matrix, then it holds that [16]:

$$(\hat{I} - \hat{\Delta}^\top)\vec{x} = [1\ 0\ \cdots 0]^\top. \tag{2}$$



(a) An SC-net, in which all binding weights are assigned a value of 1.

(b) The stochastic reachability graph of SC-net in Fig. 2a

Fig. 2: An SC-net with uniform weights and its stochastic RG.

III. Stochastic Causal Nets Discovery

In this section, we begin by introducing stochastic C-nets, a novel formalism for modeling stochastic processes, followed by a discussion of their execution semantics. Subsequently, we propose a stochastic discovery technique to extend a C-net with stochastic information derived from event log. The stochastic discovery problem is formulated as an optimization problem that maximizes the uEMSC metric between the Markovian abstraction of the stochastic C-net and log.

A. Stochastic Causal Nets

Definition 11 (Stochastic Causal Nets)**.** A stochastic causal net (SC-net) is a tuple $(A, D, I, O, a_i, a_f, w_i, w_o)$ where (A, D, I, O, a_i, a_f) is a C-net, $w_i \colon I \rightarrow \mathbb{R}_{>0}$ is a weight value function that maps each input binding in I to a weight value, and $w_o \colon O \rightarrow \mathbb{R}_{>0}$ is a weight value function that maps each output binding in O to a weight value.

SC-net introduces two weight functions for input and output bindings. We define the binding weights to be positive real numbers to ensure that they remain eligible for execution. For instance, the input and output bindings of the SC-net in Fig. 2a are all assigned a value of one.

Weights define the likelihood that an enacted activity binding will be executed by establishing a probability distribution over enacted activity bindings at a given state. This involves two consecutive decisions: first, which input binding to execute, and second, which output binding to execute. They together decide the probability of an enacted activity binding.

Definition 12 (Execution Probability for Input Bindings)**.** Let $N_s = (A,\ D,\ I,\ O, a_i, a_f, w_i, w_o)$ be an SC-net. The probability of executing an input binding is a function that takes a state s of N_s and as_I for activity a:

$$\mathrm{p}_{N_s}(as_I \mid s) = \begin{cases} \frac{w_i(as_I)}{\sum_{as'_I \in I_s} w_i(as'_I)} & \text{if } as_I \in I_s \\ 0 & \text{otherwise.} \end{cases}$$

The probability of executing an input binding is the ratio between its weight and the sum of the weights of all enacted input bindings. Similarly, at state $q_1 = [(b, c), (b, d)]$ for the example SC-net, the probability of input binding (c,{b}) is $1/1{+}1{+}1 = 1/3$.

As for an output binding $as_O \in O(a)$ for activity a, its probability is defined as:

$$\mathrm{p}_{N_s}(as_O) = \frac{w_o(as_O)}{\sum_{as'_O \in O(a)} w_o(as'_O)}.$$

Given the activity, the probability of each output binding is obtained as the ratio between its weight and the sum of the weights of all output bindings. It involves a local decision on which output binding to execute. In Fig. 2a, $O(b) = \{\{c\}, \{d\}, \{c, d\}\}$ is the set of output bindings for activity b, we have $\mathrm{p}(b, [c]) = \mathrm{p}(b, [d]) = \mathrm{p}(b, [c, d]) = 1/(1{+}1{+}1){=}1/3$.

Then, combining the dual execution probability resulting from input and output bindings, the execution probability of an activity binding is defined as follows.

Definition 13 (Execution Probability for Activity Bindings)**.** Let $N_s = (A, D, I, O, a_i, a_f, w_i, w_o)$ be an SC-net. The probability of executing an activity binding $b = (a, as_I, as_O)$ at state s, denoted as $\mathrm{p}_{N_s}(b \mid s)$, is defined as:

$$\mathrm{p}_{N_s}(b \mid s) = \begin{cases} \mathrm{p}_{N_s}(as_I \mid s) \cdot \mathrm{p}_{N_s}(as_O) & \text{if } b \in B_s \\ 0 & \text{otherwise.} \end{cases}$$

The probability of a binding sequence $\gamma = \langle b_0, \ldots, b_n \rangle$ in N_s is $\mathrm{p}_{N_s}(\gamma) = \prod_{1 \le i \le n} \mathrm{p}_{N_s}(b_i \mid s_i)$ where $s_i = \psi(\langle b_0, \ldots, b_{i-1} \rangle)$. To define the stochastic execution semantics of an SC-net, we apply Definition 13 to enrich its reachability graph with a transition probability function.

For instance, Fig. 2b illustrates the stochastic reachability graph of the example SC-net. The probability of path γ_1 is $1/3 \cdot 1 \cdot 1 = 1/3$, which indicates that trace $\langle b, c, e \rangle$ is $1/3$. The probability of other binding sequences can be computed in a similar way, after which the probability of each trace can be determined. Therefore, the stochastic language of the example SC-net is $[\langle b, c, e \rangle^{1/3}, \langle b, d, e \rangle^{1/3}, \langle b, c, d, e \rangle^{1/36}, \langle b, d, c, e \rangle^{1/36}, \langle b, c, e, d, e \rangle^{1/12}, \langle b, c, d, e, e \rangle^{1/18}, \langle b, d, c, e, e \rangle^{1/18}, \langle b, d, e, c, e \rangle^{1/12}]$, which is notably different from the stochastic language of the example event log L.

IV. Discovering Stochastic Causal Nets

Given an event log L and a C-net N, we construct an SC-net N_s using the control flow of N, and N_s is capable of reproducing the probability of the observed process in L. The SC-net discovery problem is formalized as follows.

Definition 14 (Discovery of SC-net with Optimized uEMSC^k)**.** Let L be an event log, N be a C-net, and $k \in \mathbb{N}_{\ge 2}$.

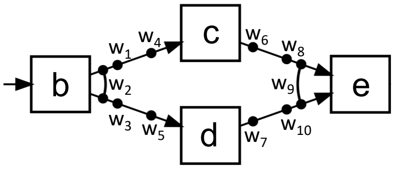


Fig. 3: The parametrized SC-net N_s.

Algorithm 1 SC-net discovery with Markovian abstraction

```
Require: $N = (A, D, I, O, a_i, a_f)$ be a C-net, $L$ be an event log,
    and $k$ be a user-defined subtrace length.
Ensure: An SC-net $N_s$ that maximizes the uEMSC$^k$ with $L$
 1: $m_L^k \leftarrow L$    ▷ Get stochastic Markovian abstraction from $L$
 2: $N_s \leftarrow (A, D, I, O, a_i, a_f, w_i, w_o)$    ▷ Initialize $N_s$
 3: $RG_s, \hat{\Delta} \leftarrow N_s$    ▷ Get stochastic RG and transition matrix
 4: $\vec{x} \leftarrow [10\cdots0] \cdot (\hat{I} - \hat{\Delta}^\top)^{-1}$    ▷ Get state probability vector
 5: for all $\gamma \in A^*$ s.t. $m_L^k(\gamma) > 0$ do
 6:     $f_{N_s}^k(\gamma) = 0$    ▷ Initialize the frequency for subtrace $\gamma$
 7:     for all $x_i$ in $\vec{x}$ do    ▷ Iterate all elements in $\vec{x}$
 8:         if $\delta(\gamma|q_i) > 0$ then    ▷ If $\gamma$ is replayable from $q_i$
 9:             $f_{N_s}^k(\gamma) \mathrel{+}= x_i \cdot \delta(\gamma|q_i)$    ▷ Sum $\gamma$'s frequency
10:         end if
11:     end for
12: end for
13: $f_{sum} \leftarrow 0$    ▷ Initialize the frequency sum of all subtraces
14: for all $x_j$ in $\vec{x}$ do    ▷ Iterate all states in $\vec{x}$
15:     $f_{sum} \mathrel{+}= \sum_{\gamma' \in A^*} x_j \cdot \delta(\gamma' \mid q)$
16: end for
17: uEMSC$^k(L, N_s) \leftarrow 1$
18: for all $\gamma \in A^*$ s.t. $f_{N_s}^k(\gamma) > 0$ do
19:     $m_{N_s}^k(\gamma) \leftarrow f_{N_s}^k(\gamma)/f_{sum}$    ▷ Normalization with $f_{sum}$
20:     uEMSC$^k(L, N_s) \mathrel{-}= \max(m_L^k(\gamma) - m_{N_s}^k(\gamma), 0)$
21: end for
22: return $N_s$ s.t. uEMSC$^k(L,N_s) = \max_{N_s' \in \mathcal{N}}$ uEMSC$(m_L^k, m_{N_s'}^k)$
```

The *stochastic SC-net discovery* problem is to find an SC-net $N_s = (A, D, I, O, a_i, a_f, w_i, w_o)$ from a set of SC-nets $\mathcal{N} = \{N_s' \mid N_s' = (A, D, I, O, a_i, a_f, w_i', w_o')\}$ with $N = (A, D, I, O, a_i, a_f)\}$, such that:

$$\forall_{N_s' \in \mathcal{N}} \ \text{uEMSC}^k(L, N_s) \geq \text{uEMSC}^k(L, N_s').$$

Definition 14 specifies that the problem involves finding an SC-net N_s from a set of candidate models such that N_s maximizes the stochastic conformance uEMSC^k with L. For instance, Fig. 3 shows an SC-net in which the values of binding weights are represented as parameters w_1, w_2, etc. Given infinitely many possible weight assignments, the goal is to find one optimal solution to weight parameters, such that uEMSC^k is maximized. This is achieved by transforming stochastic discovery into an optimization problem built on Markovian abstraction for log and model, as summarized in Algorithm 1.

Lines 1 to 4 initialize the stochastic discovery problem. First, we derive the k-th-order Markovian abstraction m_L^k for log L following Definition 8. Lines 2 and 3 establish a parametrized SC-net using the same control flow as N, initialize w_i and w_o with parametrized weights, then construct its parametrized stochastic reachability graph. Line 4 derives the vector $\vec{x}$ representing the probability of reaching each state by solving the parametrized matrix product defined in Eq. (2). Subsequently, the loop (lines 5 to 12) leverages $\vec{x}$ to compute the parametrized frequency of each subtrace in m_L^k according to model, denoted as $f_{N_s}^k(\gamma) = \sum_{q \in Q} x_q \cdot \delta(\gamma \mid q)$. Line 13 initializes f_{sum}, so that the algorithm iterates over all states to derive an overall subtrace frequency for normalization (lines 14 to 16). Lines 17 to 21 aggregate the parametrized description of uEMSC^k for event log and model. For every subtrace γ with $m_L^k(\gamma) > 0$, we extract a parametrized formula that describes the probability of γ according to the model. Finally, the parametrized uEMSC^k serves as the objective function and is maximized in line 22, returning an SC-net that achieves the maximized uEMSC^k.

For instance, for event log L and $k = 2$, we first construct 2-th-order Markovian abstraction m_L^2 following Definition 8. Then, consider SC-nets N_s in Fig. 3, in which the binding weights with parameters are represented using w_1 to w_{10}. The parametrized stochastic reachability graph of N_s is constructed, from which its parametrized transition matrix is computed. By solving the parametrized matrix equation in Eq. (2), the solution vector$\vec{x}$ of the stochastic reachability graph is constructed, as shown in Fig. 4. The element in i-th row represents the parametrized probability of reaching the i-th state of N_s, e.g., the probability of visiting q_0 is 1, q_1 is $w_2/(w_1+w_2+w_3)$, etc.

Consider $\langle b, c \rangle$ from m_L^2, we compute its frequency according to N_s by replaying $\langle b, c \rangle$ from each state in the stochastic reachability graph, and it holds that:

$$\delta(\langle b, c \rangle \mid q_i) = \begin{cases} \frac{w_2 \cdot w_4}{(w_1+w_2+w_3)(w_4+w_5)} + \frac{w_2}{(w_1+w_2+w_3)} & i = 0 \\ 0 & i \neq 0. \end{cases}$$

The frequency of $\langle b, c \rangle$ in N_s is $f_{N_s}^2(\langle b, c \rangle) = x(0) \cdot \delta(\langle b, c \rangle \mid q_0)$, where $x(0) = 1$ according to Fig. 4. Thus, $f_{N_s}^2(\langle b, c \rangle) = \frac{w_2 \cdot w_4}{(w_1+w_2+w_3)(w_4+w_5)} + \frac{w_2}{(w_1+w_2+w_3)}$. Then, $m_{N_s}^2(\langle b, c \rangle)$ is computed by normalizing $f_{N_s}^2(\langle b, c \rangle)$ with the sum of all subtrace frequencies f_{sum}.

Likewise, for other subtraces of length 2 from m_L^2, we compute their parametrized frequency according to model, after which a normalization is applied to compute their parametrized probabilities. Finally, we aggregate them and construct an objective function of uEMSC$^2(L, N_s)$, which substitutes each subtrace probability for $m_{N_s}^2$ with the parametrized representation. A solution that maximizes uEMSC$^2(L, N_s)$ leads to an SC-net that satisfies Definition 14.

Stochastic discovery using sampling Alternatively, we propose a sampling approach to approximate the parametrized Markovian abstraction of SC-net. We unfold the model using breadth-first search (BFS) and sample binding sequences up to a user-defined threshold. During sampling, we track the probability of each binding sequence. Although BFS-based sampling inherently favors shorter sequences, this bias aligns with realistic process behavior, since longer binding sequences traverse more decision points and are statistically less likely to occur in reality. After projecting the sampled binding sequences to a finite sample of traces, their corresponding

$$\vec{x} = \begin{pmatrix} 1 \\ \frac{w_2}{w_1+w_2+w_3} \\ \frac{w_2 w_5}{(w_1+w_2+w_3)(w_4+w_5)} \\ \frac{w_2 w_4}{(w_1+w_2+w_3)(w_4+w_5)} \\ \frac{w_1(w_{10}+w_4)(w_4+w_5)+w_{10}w_2w_5}{w_1(w_{10}+w_4)(w_4+w_5)+w_2w_5w_{10}} \\ \frac{w_2w_4w_5(1+w_{10}+w_5+w_8))}{(w_1+w_2+w_3)(w_{10}+w_4)(w_4+w_5)(w_5+w_8)} \\ \frac{w_2w_4w_8+w_3(w_4+w_5)(w_5+w_8)}{(w_1+w_2+w_3)(w_4+w_5)(w_5+w_8)} \\ \frac{w_1(w_{10}+w_4)(w_4+w_5)(w_5+w_8)(w_8+w_9)+w_2w_5(w_{10}+w_4)(w_{10}+w_8)(w_5+w_8)}{w_1(w_{10}+w_4)(w_5+w_8)(w_{10}(w_4+w_5)+w_4(w_5+w_9))+w_{10}^2(w_2+w_3)(w_4+w_5)(w_5+w_8)+(w_2+w_3)(w_4+w_5)(w_5+w_8)(w_4w_8+w_8w_9+w_5w_9)} \\ \frac{w_{10}^2(w_5+w_8)(w_4(w_2+w_3)+w_3w_5)+w_{10}(w_5+w_8)((w_2+w_3)w_4(w_4+w_5)+w_3(w_5w_9+w_8w_9+w_4w_8))+(w_2+w_3)w_4(w_4+w_5)(w_5+w_8)(w_8+w_9)}{w_1(w_{10}+w_4)(w_5+w_8)(w_{10}(w_4+w_5)+w_4(w_5+w_9))+w_{10}^2(w_2+w_3)(w_4+w_5)(w_5+w_8)+(w_2+w_3)(w_4+w_5)(w_5+w_8)(w_4w_8+w_8w_9+w_5w_9)} \\ 1 \end{pmatrix}$$

Fig. 4: The solution vector that represents the parametrized probability of visiting each state.

TABLE I: The uEMSC^k for event log L and SC-nets.

SC-net	uEMSC^k		
	$k = 2$	$k = 3$	$k \geq 4$
uniform binding weights	0.831	0.604	0.297
optimized binding weights	≈ 1	≈ 1	≈ 1

parametrized probabilities according to model are used to construct a stochastic language.

Subsequently, the parametrized Markovian abstraction of the model's stochastic language using Definition 8. Similarly, we compute the Markovian abstraction for the event log, where each subtrace is compared against the corresponding parametrized formula from the model's Markovian abstraction. This comparison allows for the construction of the objective function uEMSC^k between the event log and the model. Finally, stochastic C-net discovery is performed by maximizing the uEMSC^k between the model and log.

Example & Implementation With respect to log L, one optimal weights assignment for the SC-net in Fig. 3 is $w_1 = w_3 = 1, w_2 = w_5 = 8, w_4 = 3, w_6 = w_7 = w_9 = 10, w_8 = w_{10} = \epsilon$ where ϵ is a marginal value with $\epsilon > 0$. Given the weights, the traces that are unobserved in L is still possible according to models but have a marginal probability. We present the result of uEMSC^k between the event log and SC-net using uniform weights and optimized weights in Table I. Compared to SC-net with optimized binding weights, SC-net that has uniform binding weights of one shows much worse uEMSC^k with respect to the event log.

The proposed Markovian-based optimization and the sampling technique have been implemented and made publicly available [1]. The technique takes a C-net, an event log, and a user-defined Markovian order as input. The parametrized matrix equation for Eq. (2) is algebraically solved with the SymPy library [15]. Then, the objective function for parametrized uEMSC^k is constructed, and subsequently a maximized result is computed with "Basin-hopping Optimizer" from the SciPy library [17]. The optimizer progresses iteratively by refining the weights of bindings until convergence, and the algorithm return an SC-net using the derived weights.

V. Evaluation

To evaluate the feasibility of the approach, we conduct experiments with several publicly available event logs.[2] For each event log, we discover one C-net with Flexible Heuristic Miner (FHM) [18], and the other with Fodina Miner (FM) [5]. We first examine the stochastic quality of the discovered SC-nets, and then investigate the scalability of the technique. All experiments were performed on a MacBook Pro with an M2 Pro processor, 32 GB of memory, and macOS Sequoia 15.

A. Stochastic Quality of the SC-nets

We set the target subtrace length to 2 and 3, and perform stochastic C-net discovery with 2nd and 3rd-order Markovian abstraction of log and C-net. To establish the baseline, we assign uniform binding weights to each SC-net. Then, we use the proposed optimization-based stochastic discovery techniques to construct SC-net using the input event log and C-net. For the sampling-based approach, we set the target number of sampled traces to 100 at maximum. During stochastic discovery, a 10-minute timeout was applied. After constructing the SC-nets, we perform stochastic conformance with three measures, including uEMSC^k, uEMSC, and Jensen-Shannon stochastic conformance (JSSC) [12].

The results of the stochastic conformance measures between the discovered SC-net and log are illustrated in Table II and Table III. In the table, column *Uniform* represents SC-nets with uniform weights, Dis_o represents the approach described in Algorithm 1, and Dis_s is the sampling-based optimization approach. The Dis_o technique had a timeout in 2 logs, and only the sampling-based technique produced SC-nets. This is

[1] https://github.com/brucelit/stochastic_cn

[2] https://www.tf-pm.org/resources/logs

TABLE II: Stochastic conformance of SC-nets discovered with the *2nd-order* Markovian abstraction for logs and C-nets.

		C-nets mined from FHM			C-nets mined from FM		
Event log	Measure	*Uniform*	*Dis-o*	*Dis-s*	*Uniform*	*Dis-o*	*Dis-s*
Road	uEMSC^2	0.751	0.928	0.927	0.535	0.939	0.932
	uEMSC	0.611	0.820	0.792	0.189	0.793	0.786
	JSSC	0.463	0.637	0.576	0.171	0.657	0.644
Sepsis	uEMSC^2	0.345	–	0.387	0.390	–	0.405
	uEMSC	0.002	–	0.002	0.033	–	0.041
	JSSC	0.003	–	0.001	0.025	–	0.019
BPIC2017 Application	uEMSC^2	0.817	0.948	0.935	0.563	0.904	0.889
	uEMSC	0.660	0.798	0.821	0.037	0.478	0.473
	JSSC	0.593	0.748	0.742	0.042	0.343	0.339
BPIC2020 International	uEMSC^2	0.511	–	0.861	0.395	–	0.655
	uEMSC	0.107	–	0.385	0.005	–	0.125
	JSSC	0.110	–	0.349	0.007	–	0.106

TABLE III: Stochastic conformance of SC-nets discovered with the *3rd-order* Markovian abstraction of logs and C-nets.

		C-nets mined from FHM			C-nets mined from FM		
Event log	Measure	*Uniform*	*Dis-o*	*Dis-s*	*Uniform*	*Dis-o*	*Dis-s*
Road	uEMSC^3	0.625	0.912	0.882	0.302	0.892	0.878
	uEMSC	0.611	0.820	0.794	0.189	0.824	0.819
	JSSC	0.463	0.591	0.567	0.171	0.684	0.662
Sepsis	uEMSC^3	0.237	–	0.273	0.231	–	0.242
	uEMSC	0.002	–	0.002	0.033	–	0.036
	JSSC	0.003	–	0.002	0.025	–	0.021
BPIC2017 Application	uEMSC^3	0.782	0.931	0.923	0.438	0.872	0.866
	uEMSC	0.660	0.802	0.820	0.037	0.518	0.502
	JSSC	0.593	0.750	0.746	0.042	0.395	0.374
BPIC2020 International	uEMSC^3	0.411	–	0.499	0.264	–	0.596
	uEMSC	0.107	–	0.214	0.005	–	0.149
	JSSC	0.110	–	0.186	0.007	–	0.112

attributed to the large number of states that generate a large, sparse transition matrix. To derive the vector with parametrized representations of each state, the technique cannot solve it efficiently due to the large number of parameters.

For SC-nets derived from the Road and Application logs, the sampling technique effectively approximates the optimization technique, as evidenced by the close stochastic conformance results. However, for the Sepsis log, the stochastic difference between baseline models and SC-nets discovered through sampling is relatively marginal. This occurs because the underlying control-flow model has low fitness, which adversely affects the stochastic quality of the discovered models.

Overall, compared to SC-nets with uniform weights, the SC-nets constructed using the proposed approach show a superior stochastic conformance. In particular, for C-nets mined from FM, the SC-nets mined from the proposed approach achieve a significant stochastic conformance improvement compared to baseline models. Since no other algorithms for discovering SC-nets exist, we cannot compare our performance directly against other implementations.

B. *Scalability*

Our stochastic discovery approach involves a hyperparameter, the order of Markovian abstraction k, which accounts for subtraces of length k in both the event log and the SC-net. We examined the scalability of our technique using values of k ranging from 2 to 4. During stochastic discovery, we applied a 10-minute timeout, and the sampling-based technique samples up to 100 traces from the model.

Figure 5 illustrates the run time with optimization- and sampling-based discovery techniques. For discoverable SC-nets, the running time increases linearly with respect to the order of Markovian abstraction due to the growing complexity of the parameterized objective function. However, the optimization-based discovery approach failed to construct SC-nets for the International and Sepsis event logs within the time limit, which stems from the computational complexity of the parametrized matrix product. For a C-net of large size, we recommend using sampling-based technique.

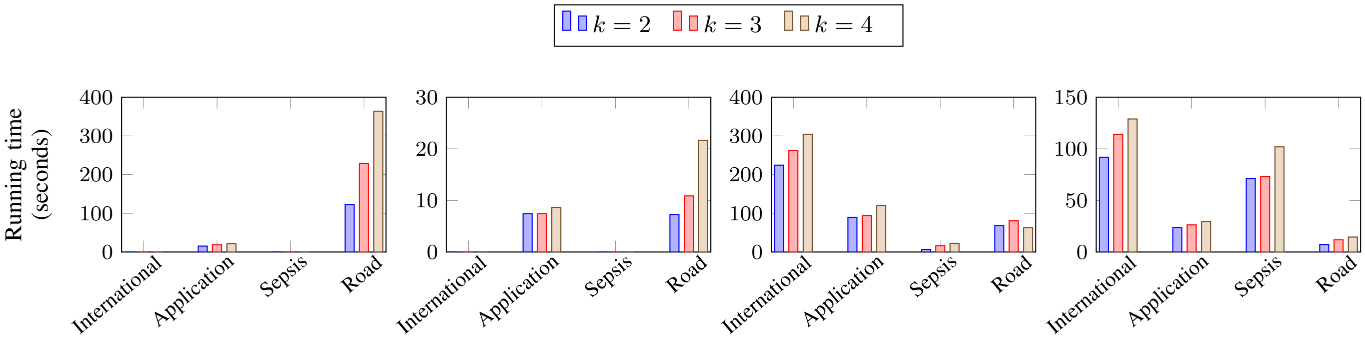
VI. Related Work

Existing stochastic process discovery techniques can be categorized into two types: single-stage approaches and two-stage approaches [9]. Single-stage approaches directly construct stochastic process models from event logs. Two-stage approaches first apply a standard algorithm to obtain a control flow model and then convert it into a stochastic model based on the trace probabilities of log.

Toothpaste Miner [7] is a single-stage technique capable of automatically constructing stochastic models. It applies a set of reduction and abstraction rules to reduce the size of the model and generate a probabilistic process tree as an intermediate output, then a transformation is applied to translate it into a stochastic labeled Petri net. GASPD [3] is a technique based on grammatical inference, which discovers a family of stochastic directed action graphs from an input log.

The two-stage discovery accounts for the stochastic perspective such that the probability over traces in the discovered stochastic model resembles that in the event logs. The work in [6] introduced six weight estimators based on statistics computed on event log and model. The approach in [9] optimizes the weights of transitions with respect to Entropic Relevance and uEMSC. In [8], the authors use a general, gradient-free function minimization method to maximize the Earth Movers' Stochastic Conformance (EMSC) between the constructed SPN and event log. Moreover, [4] explicitly utilizes information from the solution of EMSC by computing subgradients.

Compared to existing techniques [4, 8, 9] that transform stochastic discovery into an optimization problem for stochastic labeled Petri nets, our work differs in two key respects. First, our model is based on a stochastic extension of C-nets, a declarative modeling formalism used by several discovery algorithms. Second, we perform Markovian-based abstraction for both event log and model, thus the stochastic discovery relies on a novel stochastic conformance that accounts for the relative occurrences of subtraces and partial matches.



(a) Dis_o with C-nets mined from FM.

(b) Dis_o with C-nets mined from FHM.

(c) Dis_s with C-nets mined from FM.

(d) Dis_s with C-nets mined from FHM.

Fig. 5: The running time of the SC-net discovery using Markovian abstractions of orders 2, 3 and 4 for logs and C-nets.

VII. Conclusion

C-nets are suitable for process mining given their expressiveness without including different additional model elements such as silent transitions, places, or gateways [2]. In this paper, we have extended its semantics in stochastic settings to model not only the control flow of the process but also the probability distribution over its traces.

Given an event log and a C-net, we proposed a technique to discover a SC-net with optimized weights of bindings such that a stochastic conformance measure regarding the event log is maximized. In particular, the stochastic conformance measure we employ explicitly considers partial matching of subtraces, as traces may deviate from the model while still representing meaningful operational patterns. This approach complements existing stochastic discovery techniques that treat any trace with zero probability under the model as entirely incompatible, thereby including more granular behavior for stochastic discovery.

For C-nets with large state spaces where direct optimization is computationally infeasible, we proposed a sampling-based technique that approximates the model's stochastic behavior. Our evaluation also confirms that the discovered SC-nets show superior stochastic quality compared to baseline models.

Several directions exist for future research. For instance, one could study the representational bias of stochastic C-net and identify a mapping from stochastic C-net to stochastic labeled Petri nets (SLPNs), as existing stochastic process mining techniques are designed mainly for SLPNs. This work motivates future research on SC-nets for stochastic conformance checking [11, 13]. Moreover, one can investigate other stochastic conformance measures and a single-stage approach that directly constructs an SC-net from the event log.

References

[1] van der Aalst, W.M.P.: Process Mining - Data Science in Action, Second Edition. Springer (2016)

[2] van der Aalst, W.M.P., Adriansyah, A., van Dongen, B.F.: Causal nets: A modeling language tailored towards process discovery. In: CONCUR. LNCS, vol. 6901, pp. 28–42. Springer (2011)

[3] Alkhammash, H., Polyvyanyy, A., Moffat, A.: Stochastic directly-follows process discovery using grammatical inference. In: CAiSE. LNCS, vol. 14663, pp. 87–103. Springer (2024)

[4] Brockhoff, T., Uysal, M.S., van der Aalst, W.M.P.: Wasserstein weight estimation for stochastic petri nets. In: ICPM. pp. 81–88. IEEE (2024)

[5] vanden Broucke, S.K.L.M., Weerdt, J.D.: Fodina: A robust and flexible heuristic process discovery technique. Decis. Support Syst. **100**, 109–118 (2017)

[6] Burke, A., Leemans, S.J.J., Wynn, M.T.: Stochastic process discovery by weight estimation. In: ICPM Workshops. LNBIP, vol. 406, pp. 260–272. Springer (2020)

[7] Burke, A., Leemans, S.J.J., Wynn, M.T.: Discovering stochastic process models by reduction and abstraction. In: Petri Nets. LNCS, vol. 12734, pp. 312–336. Springer (2021)

[8] Cry, P., Horváth, A., Ballarini, P., Gall, P.L.: A framework for optimisation based stochastic process discovery. In: QEST+FORMATS. LNCS, vol. 14996, pp. 34–51. Springer (2024)

[9] Leemans, S.J.J., Li, T., Montali, M., Polyvyanyy, A.: Stochastic process discovery: Can it be done optimally? In: CAiSE. LNCS, vol. 14663, pp. 36–52. Springer (2024)

[10] Leemans, S.J.J., Polyvyanyy, A.: Stochastic-aware precision and recall measures for conformance checking in process mining. Inf. Syst. **115**, 102197 (2023)

[11] Leemans, S.J.J., Syring, A.F., van der Aalst, W.M.P.: Earth movers' stochastic conformance checking. In: BPM Forum. LNBIP, vol. 360, pp. 127–143. Springer (2019)

[12] Li, T., Leemans, S.J.J., Polyvyanyy, A.: The jensen-shannon distance for stochastic conformance checking. In: ICPM Workshops. LNBIP, vol. 533, pp. 70–83. Springer (2024)

[13] Li, T., Polyvyanyy, A., Leemans, S.J.J.: Stochastic alignments: Matching an observed trace to stochastic process models (in press)

[14] Mannhardt, F., de Leoni, M., Reijers, H.A.: Heuristic mining revamped: An interactive, data-aware, and conformance-aware miner. In: BPM (Demos). CEUR Workshop Proceedings, vol. 1920. CEUR-WS.org (2017)

[15] Meurer, A., et al.: Sympy: symbolic computing in python. PeerJ Computer Science **3**, e103 (2017)

[16] Rocha, E.G., Leemans, S.J.J., van der Aalst, W.M.P.: Stochastic conformance checking based on expected subtrace frequency. In: ICPM. pp. 73–80. IEEE (2024)

[17] Virtanen, P., et al.: Scipy 1.0: fundamental algorithms for scientific computing in python. Nature methods **17**(3), 261–272 (2020)

[18] Weijters, A.J.M.M., Ribeiro, J.T.S.: Flexible heuristics miner (FHM). In: CIDM. pp. 310–317. IEEE (2011)