Stochastic Process Discovery: Can it be Done Optimally?

Sander J.J. Leemans[1,2], Tian Li[1,4], Marco Montali[3⋆], and Artem Polyvyanyy[4]

[1] RWTH Aachen University, Germany
{s.leemans, t.li}@bpm.rwth-aachen.de
[2] Fraunhofer, Germany
[3] Free University of Bozen-Bolzano, Italy
[4] The University of Melbourne, Australia

Abstract. Process discovery is the problem of automatically constructing a process model from an event log of an information system that supports the execution of a business process in an organisation. In this paper, we study how to construct models that, in addition to the control flow of the process, capture the importance, in terms of probabilities, of various execution scenarios of the process. Such probabilistic aspects of the process are instrumental in understanding the process and to predict aspects of its future. We formally define the problem of stochastic process discovery, which aims to describe the processes captured in the event log. We study several implications of this definition, and introduce two discovery techniques that return optimal solutions in the presence and absence of a model of the control flow of the process. The proposed discovery techniques have been implemented and are publicly available. Finally, we evaluate the feasibility and applicability of the new techniques and show that their models outperform models constructed using existing stochastic discovery techniques.

Keywords: Stochastic process mining, stochastic process discovery

1 Introduction

The increasing complexity of modern socio-technical and cyber-physical systems is calling for a change of paradigm in their engineering, moving from pure model-driven engineering to approaches where *models and execution data are synergically connected* [16]. In enterprise information systems engineering, this practice has a long tradition when focusing on *how* organisations operate, that is, on their work processes. Specifically, process mining techniques [1] provide insights on processes by analysing the event data produced within an organisation while executing such processes. Even data yield, implicitly or explicitly, *event logs* recording the historical executions of the process under scrutiny, where

⋆ This work is partially supported by the UNIBZ project ADAPTERS and the PRIN MIUR project PINPOINT Prot. 2020FNEB27.

each execution typically corresponds, in the log, to an (execution) *trace*: a time-ordered sequence of triggered events, each referring to an activity within the process. A key process mining task is that of *process discovery*, whose goal is to learn a process model that suitably reconstructs the behaviours contained in the event log under scrutiny, and that can be used as a driver for fact-based process analysis and improvement. However, traditional process discovery techniques do not transfer into the discovered models any information regarding the relative frequency, and in turn the likelihood, of the observed traces. This hampers the possibility of using such models, as letting infrequent flows in the process influence optimisation and analysis indistinctly from frequent ones is improvident.

To tackle this limitation, *stochastic process discovery* techniques learn process models that pair traces with indications on how likely one can expect to see them in the future executions of the process. This is challenging due to a granularity mismatch in the stochastic information contained in a log or model. At the log level, the likelihood of a trace is directly obtained by dividing its frequency with the total number of log traces. At the model level, stochastic information is usually attached locally to decision points, and the likelihood of a trace is only indirectly obtained by chaining such (independent) decisions. In spite of this challenge, several stochastic process discovery techniques have been proposed [19,7,6], empirically demonstrating their applicability and quality using real-life logs. On the downside, none of these works provides foundational insights on the problem space and on the formal properties of the proposed techniques, in terms of optimality and guarantees.

The goal of this work is to fill this gap, providing a *foundational investigation of stochastic process discovery*, in the case where the target process model is expressed using stochastic variants of Petri nets [17,15,11]. We provide a formal definition of the problem, casting it as a two-dimensional optimisation problem. The first dimension concerns the *behaviour* of the process, that is, the selection of a process model in a class of Petri nets defined based on some representational bias defining which constructs can be expressed (e.g., whether labels can be repeated). The second dimension concerns the *stochastic information* attached to the process, employing measures that define how well the distribution of traces induced by the model matches the distribution of the log. We study the implications of this definition in terms of optimality, considering two types of stochastic process discovery techniques. The techniques from the first type operate under the assumption that the control flow of the target process model is given, and the goal is to enhance it with the "best" stochastic information. The techniques of the second type relax this assumption, and entangle control-flow and stochastic information discovery in a single step. Finally, we evaluate the introduced techniques in the context of existing stochastic process discovery algorithms.

The paper is organised as follows. In Section 2, we discuss related work, while in Section 3, we introduce existing concepts. Section 4 formally introduces the problem of stochastic process discovery and discusses some inherent aspects of this definition. Sections 5 and 6 introduce two discovery techniques. Section 7 evaluates the introduced techniques before Section 8 concludes the paper.

2 Related Work

The representation, discovery and measurements of stochastic process models have been investigated before in several settings, leading to a variety of techniques that deal with different prerequisites, targets, advantages and limitations.

Several formalisms are used in process mining for representing stochastic process models. Generalised Stochastic Petri nets are a well-established formalism for stochastic modelling [3]. Molloy [17] introduced the first model to handle the stochastic aspect, which was extended later by Marsan et al. [15] by distinguishing timed and immediate transitions, showing how the resulting stochastic behaviour can be captured through a discrete-time Markov chain.

Several measures have been proposed to quantify the quality of a stochastic process model with respect to an event log. Entropic relevance measures the average number of bits required to compress a trace in the input event log based on the structure and information about the relative likelihoods of traces provided by the stochastic process model [18]. Consequently, a model with a lower relevance value to a given log is accepted as such that describes the traces and their likelihoods better. Earth Movers' Stochastic Conformance (EMSC) derives the stochastic languages from the input event log and model and then measures the earth movers' distance between them [14].

The ability to discover the stochastic perspective of processes has enabled new types of analyses in process mining, such as analysis tasks on the traces of labelled stochastic processes and their probabilities [11]—instrumental to provide exact methods for computing stochastic conformance measures, detection of stochastic-based changes in processes [5], and techniques for weighting alignments depending on the likelihood of model traces [4].

Existing stochastic process discovery techniques can be classified into two categories: *one-stage* techniques directly discovering a stochastic model from an event log and *two-stage* approaches that first discover a control-flow model, and then annotate it with stochastic information. To the best of our knowledge, Toothpaste Miner [7] is the only known stochastic discovery technique that automatically outputs a stochastic process model without relying on a given control flow model. However, this approach does not provide conformance-measure guarantees. Recently, several two-stage approaches have been proposed. The GDT_SPN Miner [19] is an alignment-based technique that estimates arbitrary delay distributions of stochastic Petri nets. The weight estimation framework includes deterministic and non-deterministic estimators that derive an SPN [6]. In [12], the authors proposed the discovery of stochastic dependencies, which address the likelihood of decisions influenced by earlier decisions. Two-stage discovery is reminiscent of the widely investigated problem of parameter synthesis in probabilistic models (in particular, Markov chains) [8]. The crucial difference is that in parameter synthesis there is a single probabilistic reachability/temporal property used to drive the search for parameter assignments, while in two-stage discovery this is done by considering multiple properties, expressing that the probability of each log trace according to the model should resemble the frequency with which it appears in the log. To the best of our knowledge, the limi-

tations of stochastic process discovery have not been studied, and no stochastic process discovery technique that guarantees some kind of optimal result with respect to a conformance measure have been proposed.

3 Preliminaries

Multisets, logs, vectors. A multiset $X\colon S \to \mathbb{N}$ is a mapping of elements in S to the natural numbers. The multiset union is $(X_1 \uplus X_2)(a) \equiv X_1(a) + X_2(a)$, the multiset subset is $X_2 \subseteq X_1 \equiv \forall_{a \in S} X_1(a) \geq X_2(a)$, and for any multiset X over S, $X \subseteq S^\infty$. For multisets X, X', if $X \subseteq X'$ then $(X \setminus X')(a) = X(a) - X'(a)$ is the multiset difference. Thus, the multiset $X = [x^2, y^3, z^5]$ contains 10 items, and $|X| = 3$ while $||X|| = 10$. $\bar{X} = \{a \mid X(a) > 0\}$ is the corresponding set.

A *trace* is a sequence of activities that denotes the process steps executed for a particular case of a process. An *event log* is a multiset of traces. Figure 1c is an event log with 100 traces. For an (event) log L, $\tilde{L}$ denotes its prefix closure: $\tilde{L} \equiv \{\langle a_1 \dots a_m \rangle \mid \langle a_1 \dots a_m, \dots \rangle \in \bar{L}\}$. The set of all logs is $\mathcal{L}$.

Let $\boldsymbol{A} = \langle a_1, \dots, a_n \rangle$, $\boldsymbol{B} = \langle b_1, \dots, b_n \rangle$ be vectors. Then, $\boldsymbol{A} \succeq \boldsymbol{B} \equiv \forall_{1 \leq i \leq n} a_i \geq b_i$, and $\boldsymbol{A} \odot \boldsymbol{B} \equiv \langle a_1 b_1, \dots, a_n b_n \rangle$; we might omit $\odot$ if the context is clear.

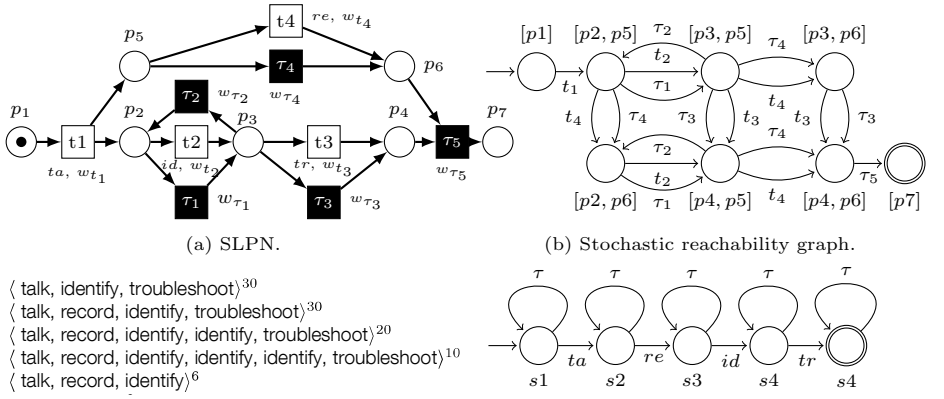
Stochastic models. A stochastic language is a weighted collection of traces, such that the sum of weights of all traces in the language is 1. A stochastic model M expresses a stochastic language. The set of all stochastic models is $\mathcal{M}$.

Definition 1 (Stochastic Petri net). *Let P be a set of places, let T be a set of transitions such that $P \cap T = \emptyset$, let $F \subseteq (P \times T \uplus T \times P)^\infty$ be a flow relation, let $w\colon T \to \mathbb{R}^+$ be a weight function and let $M_0 \subseteq P^\infty$ be an initial marking. Then, (P, T, F, w, M_0) is a* stochastic Petri net *(SPN).*

Definition 2 (Stochastic labelled Petri net). *Let Σ be an alphabet of activities, let (P, T, F, w, M_0) be an SPN and let $\lambda\colon T \to \Sigma \cup \tau$ be a labelling function. Then, (P, T, F, w, M_0) is a* stochastic labelled Petri net *(SLPN).*

An SPN or SLPN starts execution in its initial marking M_0. Let $t^\bullet = [p \mid (t, p) \in F]$ and ${}^\bullet t = [p \mid (p, t) \in F]$. In a marking M, the transitions $T_e = \{t \mid {}^\bullet t \subseteq M\}$ are *enabled*. An enabled transition $t \in T_e$ can *fire*, with firing probability $\mathbb{P}(t \mid M) = \frac{w(t)}{\Sigma_{t' \in T_e} w(t')}$, which results in a new marking $M' = M \uplus t^\bullet \setminus {}^\bullet t$. A *path* is a sequence of transitions $\langle t_1 \dots t_n \rangle$ such that there is a sequence of markings $\langle M_0 \dots M_n \rangle$ such that $\forall_{1 \leq i \leq n} {}^\bullet t_i \subseteq M_{i-1} \wedge M_i = M_{i-1} \uplus t_i^\bullet \setminus {}^\bullet t_i$ and M_n is a *deadlock*, that is, $\neg \exists_{t \in T} {}^\bullet t \subseteq M_n$. That is, a path brings the model from its initial marking M_0 to a deadlock marking. The probability of a path $\langle t_0 \dots t_n \rangle$ is $\prod_{1 \leq i \leq n} \mathbb{P}(t_i \mid M_i)$ where $M_i = M_{i-1} \uplus t^\bullet \setminus {}^\bullet t$.

A *trace* is a sequence of activities. For an SPN, path and trace are equivalent notions, while for an SLPN, the projection of a path by λ on the non-τ transitions is a trace. In an SLPN, there may be several (even countable-infinitely many [10]) paths that project to the same trace. For an SPN or SLPN M and a trace σ, we write $M(\sigma)$ for the probability of σ in M. For σ, we introduce an automaton



(a) SLPN.

(b) Stochastic reachability graph.

(c) Event log.

(d) SDFA for $\langle ta, re, id, tr \rangle$.

Fig. 1: Example log and model.

that accepts σ interleaved with arbitrary silent transitions (Figure 1d shows an example):

Definition 3 (Silenced trace DFA). *Let τ be the silent label. The* silenced trace deterministic finite automaton *(SDFA) of trace $\sigma = \langle a_1, \ldots, a_n \rangle$ is a tuple $(\Sigma, S, s_0, S_f, \delta)$ where $\Sigma = \{a_0, \ldots, a_n\}$, $S = \{s_0, \ldots, s_{n+1}\}$, $S_f = \{s_{n+1}\}$, and $\delta = \{s_i \times a_i \to s_{i+1} \text{ for } 1 \leq i < n+1\} \cup \{s_i \times \tau \to s_i \text{ for } 1 \leq i < n+1\}$.*

Definition 4 (Stochastic Reachability Graph). *The stochastic reachability graph of an SLPN M is a labelled transition system $R = (\Sigma, S, s_0, S_f, \varrho, p)$ where Σ is a finite set of labels, S is a set of states (reachable markings in M), $s_0 \in S$ is the initial state, $S_f \subseteq S$ is the set of accepting states, $\varrho \colon S \times \Sigma \to S$ is a transition function, and $p \colon \varrho \to [0, 1]$ is a probability function that maps each transition in ϱ to a probability value, such that for every transition $t = \langle s, l, s' \rangle \in \varrho$, $p(t) = \mathbb{P}(t|s)$, and for every non-deadlock marking $s \in S$, $\sum_{t=\langle s,l,s_2 \rangle \in \varrho} p(t) = 1$.*

If we disregard labels in the stochastic reachability graph and only consider firing probabilities, the graph is a discrete-time absorbing Markov chain, by mapping the final states to the absorbing states, non-final states to the transient states, and strip off transition labels, while keeping the probabilities.

Definition 5 (Absorbing Markov Chain for Stochastic Reachability Graph). *Let R be a stochastic reachability graph, its absorbing Markov chain is a tuple $C = (S, \varrho, p)$ where $S = S_t \cup S_a$, such that S_t is the set of transient states, and S_a is a set of absorbing states, $\varrho \subseteq S \times S$ is a transition relation, such that $\langle S, \varrho \rangle$ is a connected graph, and $p \colon S \times S \to [0, 1]$ is a probability function, such that for all states $s \in S_a \colon \sum_{t=\langle s,s' \rangle} p(t) = 1$.*

Stochastic conformance measures. A stochastic conformance measure δ compares an event log and a stochastic model, that is, $\delta \colon \mathcal{L} \times \mathcal{M} \to \mathbb{R}$. In

this work, we use two such measures: unit earth movers' stochastic conformance (uEMSC [14]) and inverted entropic relevance (ER^{-1} [2]).

uEMSC captures the agreement mass between the distributions of L and M.

Definition 6 (Unit Earth Movers' Stochastic Conformance [14]). *Let L be an event log and let M be an SLPN. Then, the* unit Earth Movers' Stochastic Conformance *is* $\text{uEMSC}(L, M) = 1 - \sum_{\sigma \in \bar{L}} \max(L(\sigma) - M(\sigma), 0)$.

An entropic relevance of a stochastic process model M to an event log L measures the average number of bits required to describe a trace in L given the information available in M.[5] The lower the relevance, the better the model describes the stochastic language of the log. To allow consistent discussions, in this work, we invert entropic relevance to obtain a conformance measure.

Definition 7 (Inverted Entropic Relevance [2]). *Let L be a non-empty event log and let M be an SLPN. Let Λ be the set of all activities appearing in the traces of L. Then, the* inverted entropic relevance (ER^{-1}) *of M to L is defined as follows:*

$$ER^{-1}(L, M) = \frac{1}{H_0\left(\sum_{\sigma \in \bar{L},\, M(\sigma) > 0} L(\sigma)\right) + \sum_{\sigma \in \bar{L}} L(\sigma) J(\sigma, M)}$$

$$J(\sigma, M) = \begin{cases} -\log_2 M(\sigma) & M(\sigma) > 0 \\ (1 + |\sigma|) \log_2(1 + |\Lambda|)) & \textit{otherwise} \end{cases}$$

$$H_0(x) = -x \log_2 x - (1 - x) \log_2 (1 - x) \textit{ with } H_0(0) = H_0(1) = 0$$

4 The Stochastic Discovery Problem

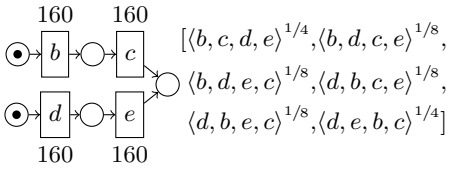
In this section, we formally define the stochastic discovery problem, show several direct implications of our definition, and prove a generic result for uEMSC.

Definition 8 (Stochastic process discovery problem). *Let L be an event log and let $\delta \colon \mathcal{L} \times \mathcal{M} \to \mathbb{R}^{\geq 0}$ be a stochastic conformance measure. Then, the* stochastic discovery problem *is to find a model M of a class of models $\mathcal{M}'$ such that M maximises stochastic conformance with L:*

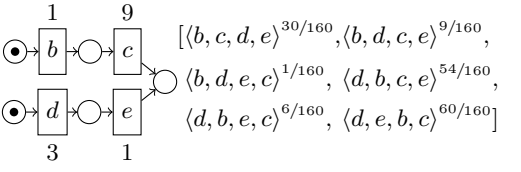
$$\delta(L, M) = \max_{M' \in \mathcal{M}'} \delta(L, M')$$

For the representational bias $\mathcal{M}'$ of all $SLPN$s, this definition is prone to over-fitting, and trivial solutions exist. For instance, for $\delta = \text{uEMSC}$ and representational bias $\mathcal{M}'$ of all SLPNs, a technique that satisfies this definition would be to return an SLPN representing every trace of L with the likelihood of that trace (a *stochastic trace model*), which would be a useless exercise as nothing new would have been learned and the model would be too complex for

[5] We use entropic relevance that relies on the uniform background coding model [2].



(a) SPN 1.



(b) SPN 2.

$[\langle b,c,d,e\rangle^{30}, \quad \langle b,d,c,e\rangle^{9}, \quad \langle b,d,e,c\rangle^{1},$
$\langle d,b,c,e\rangle^{54}, \quad \langle d,b,e,c\rangle^{6}, \quad \langle d,e,b,c\rangle^{60}]$

(c) Log L.

	SPN 1	SPN 2
ER^{-1}	0.410	0.508
EMSC	0.831	1.000
uEMSC	0.663	1.000

(d) Conformance.

Fig. 2: Example of frequencies vs. weights.

human analysis. Furthermore, in evaluation settings, to avoid over-fitting, one should either separate evaluation and training data, or consider an appropriate representational bias, and obviously the definition given here does not guarantee optimality in such cases. As such, this definition is rather limited, but allows us to discuss stochastic process discovery in more detail, and obtain techniques that nevertheless perform competitively in evaluations, even though the optimality does not extend to these evaluation settings.

4.1 Implications

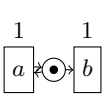
Weights vs. frequencies. Regardless of the representational bias, how often a transition is executed (its *frequency*) is not necessarily proportional to its weight in a stochastic model.

Figure 2 shows an example of two SPNs with concurrency, only differing in their stochastic perspective. In the SPNs and the log, all activities occur exactly once per trace. Thus, an estimator technique based on frequencies, such as [6], will assign equal weights to all transitions, for instance SPN 1. However, next to *which* activities are executed, the weights in an SPN also influence the *order* of activities. As such, intuitively SPN 2 is a much more likely explanation for the log than SPN 1. This is reflected in the stochastic conformance measures, shown in Figure 2d, all of which assign higher scores to SPN 2.

Another example is shown in Fig. 4. SPN 5 is frequency-based, while in SPN 6, b has twice the weight of a. The latter model has higher uEMSC and EMSC scores, as it prioritises the trace $\langle b,d\rangle$ at the expense of the a traces, which, in this case, proves beneficial for these measures. ER^{-1} of SPN 6 is lower than that of SPN 5, though. Despite traces $\langle a,d\rangle$ and $\langle b,d\rangle$ are modelled by SPN 6 perfectly, the probabilities of traces $\langle a,c\rangle$ and $\langle b,c\rangle$ as per SPN 6 deviate further from those in the log, as compared to their probabilities in SPN 5, causing non-linear effects on ER^{-1}. This example shows that frequency-based weight estimators are also challenged by dependent choices.

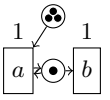
Loops. In the representational bias of SPNs, a model with loops is unlikely to be the result of stochastic process discovery according to Definition 8.

For instance, consider Figure 3. In SPN 3, $1/16$ probability mass is included in traces with 4 or more as, which does not appear in the event log. In SPN 4, the



(a) SPN 3.

$[\langle b \rangle^{1/2}, \quad \langle a, b \rangle^{1/4}, \quad \langle a, a, b \rangle^{1/8}, \quad \ldots]$



(b) SPN 4.

$[\langle b \rangle^{1/2}, \quad \langle a, b \rangle^{1/4}, \quad \langle a, a, b \rangle^{1/8}, \quad \langle a, a, a, b \rangle^{1/8}]$

$[\langle b \rangle^{8}, \quad \langle a, b \rangle^{4}, \quad \langle a, a, b \rangle^{2}, \quad \langle a, a, a, b \rangle^{1}]$

(c) Log L.

	SPN 3	SPN 4
ER^{-1}	0.577	0.600
EMSC	$\approx$ 0.958	0.965
uEMSC	0.938	0.942

(d) Conformance measures.

Fig. 3: Example of the influence of bounding loops.

loop is bounded by an extra place with 3 tokens, such that this probability mass is not "lost" on traces that are not in the event log, but instead is put on the longest trace. This is reflected in the conformance measures shown in Figure 3d.

In Lemma 1, we will prove that in general, uEMSC can only go up when adding such bounding places that do not restrict the behaviour in the log.

This example also shows the limitations of Definition 8: even though uEMSC may increase a little by adding such places, the model also gets more complex.

Two-stage approach. Regardless of the representational bias, existing stochastic process discovery approaches can be categorised by that the technique either (i) discovers control flow itself (one-stage, e.g. [7]) or (ii) leverages a control flow model as input (two-stage, e.g. [6]).

Figure 4 shows an example of the limitations of a two-stage approach, that is, first discovering a process model and then estimating the weights on top of it. In this example, the likelihoods of c and d depend on the choice between a and b. Comparing SPN 5 to SPN 7, we observe that SPN 5 (without its stochastic perspective) is a fully fitting model, but is less precise than SPN 7, whereas SPN 7 has a slightly lower fitness but a higher precision.

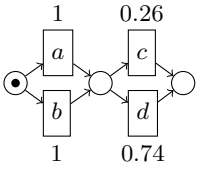
The stochastic measures, as shown in Figure 4e, clearly prefer SPN 7. Thus, a control-flow trade-off needs to be made, with implications on the stochastic perspective. Hence, in this example, the stochastic perspective needs to be considered to decide on the control flow structure of the model.

The example shows that a lower fitness does not guarantee lower stochastic quality, and a higher fitness does not guarantee a higher stochastic quality. Therefore, it may be challenging to choose a model if stochastic quality is to be optimised. Thus, a two-stage approach may not always yield the best results.

4.2 uEMSC and Precision

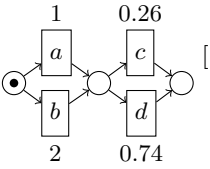
Next, we establish the relation between control-flow precision and uEMSC. That is, we show that if non-log traces are removed from a model, uEMSC can only improve (which is conceptually linked to precision in [20, A2]).

Lemma 1 (Precision - uEMSC monotonicity). *Let L be an event log, and let M and M' be SPNs, such that $M = (P, T, F, w, M_0)$ and $M' = (P', T, F', w, M_0)$,*



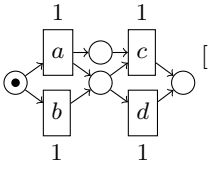
$[\langle a, c\rangle^{0.13}, \quad \langle a, d\rangle^{0.37}, \quad \langle b, c\rangle^{0.13}, \quad \langle b, d\rangle^{0.37}]$

(a) SPN 5 (frequency-based).



$[\langle a, c\rangle^{0.09}, \quad \langle a, d\rangle^{0.25}, \quad \langle b, c\rangle^{0.17}, \quad \langle b, d\rangle^{0.49}]$

(b) SPN 6 (better fit).



$[\langle a, c\rangle^{0.25}, \quad \langle a, d\rangle^{0.25}, \quad \langle b, c\rangle^{0}, \quad \langle b, d\rangle^{0.50}]$

(c) SPN 7.

$[\langle a, c\rangle^{25}, \quad \langle a, d\rangle^{25}, \quad \langle b, c\rangle^{1}, \quad \langle b, d\rangle^{49}]$

(d) Event log.

	SPN 5	SPN 6	SPN 7
ER^{-1}	0.547	0.523	0.610
EMSC	0.880	0.917	0.995
uEMSC	0.760	0.833	0.990

(e) Measures.

Fig. 4: Example of the interplay between places and weights.

such that $\tilde{L} \cap \tilde{M} = \tilde{L} \cap \tilde{M'}$*, and such that* $\tilde{M'} \subset \tilde{M}$*. Then,* $\text{uEMSC}(L, M) \leq \text{uEMSC}(L, M')$*.*

Proof. Let $\sigma \in \tilde{M}$, $\sigma \notin \tilde{M'}$, $\sigma \notin \tilde{L}$. Let $\rho = M(\sigma)$ be the probability of σ in M. As σ does not contribute to $\text{uEMSC}(L, M)$, it holds that $\text{uEMSC}(L, M) \leq 1 - \rho$. As M' is a SPN, the probability mass ρ is accounted for in traces other than σ. As some of these traces may be in L, by definition of uEMSC, $\text{uEMSC}(L, M) \leq \text{uEMSC}(L, M')$. □

Informally, ρ gets distributed from traces allowed by M that are not in the event log (that is, imprecise traces), to other traces in M'. Some traces in M' therefore get extra probability, and for some traces σ, this may increase the difference between $L(\sigma)$ and $M(\sigma)$, which leaves uEMSC unchanged as ρ already fully counted against $\text{uEMSC}(L, M)$. However, for some traces σ' it may be that this difference gets smaller, thereby increasing $\text{uEMSC}(L, M')$.

5 Stochastic Discovery as a Decision Problem

In this section, we translate stochastic process discovery with the representational bias of SPNs to a decision problem for uEMSC. That is, in this section, we consider the setting in which we do not have a control-flow model yet, and we aim to discover a fully stochastic model in one go, where we limit ourselves to the unlabelled transitions of SPNs. We start with the given set of transitions T. Then, we need to decide on four sets of variables: (i) the set of places P, (ii) the initial marking $M_0 \in P^\infty$, (iii) for every place $p \in P$ and transition $t \in T$, the arc multiplicities ${t^\bullet}_p \in \mathbb{N}$ and ${}^\bullet t_p \in \mathbb{N}$, and (iv) for every transition $t \in T$, a weight: $W_t \in (0, 1]$. For ease of notation, we write these variables as vectors, that is, $\boldsymbol{M_0}$, $\boldsymbol{t^\bullet}$, $\boldsymbol{{}^\bullet t}$ and $\boldsymbol{W}$. If $M_0(p) = {t^\bullet}_p = {}^\bullet t_p = 0$, a place p has no influence on the result, and thus the set of places P is a dependent variable and will be ignored further on.

We do not require that every trace of the log fits the model, so we need to keep track of which of the traces are supported by the model. Inspired by the ILP miner [21], if the net supports a trace σ, it should support all its prefixes, and the net should be in a deadlock after σ.

We introduce helper variables $\boldsymbol{E}$ indicating whether a pre-fix is supported by the model. As a base case, the empty pre-fix is supported by the model in any case (1), and a non-empty pre-fix $\sigma \cdot \langle t \rangle$ is supported if and only if the shorter pre-fix σ is supported and after executing all transitions in σ, t is enabled (2).

$$E_{\langle\rangle} = \text{true} \tag{1}$$

$$\forall_{\sigma \cdot \langle t \rangle \in \bar{L}} E_{\sigma \cdot \langle t \rangle} = E_\sigma \wedge \boldsymbol{M_0} + \left(\sum_{t'' \in \sigma'} (\boldsymbol{t''^\bullet} - {}^\bullet\boldsymbol{t''}) \right) \succeq {}^\bullet\boldsymbol{t} \tag{2}$$

Furthermore, we introduce helper variables $\boldsymbol{D}$ indicating whether a trace is supported by the model. That is, for every trace $\sigma \in \bar{L}$ in the log, D_σ is true if and only if the net supports σ and after σ the net is in a deadlock:

$$\forall_{\sigma \in \bar{L}} D_\sigma = E_\sigma \wedge \forall_{t \in T} \left(\boldsymbol{M_0} + \sum_{t'' \in \sigma'} \boldsymbol{t''^\bullet} - {}^\bullet\boldsymbol{t''} \right) \not\succeq {}^\bullet\boldsymbol{t} \tag{3}$$

An optimal solution for Definition 8 can then be obtained by maximising uEMSC directly in the objective function. This function multiplies the probability of a trace by whether the trace is enabled (assuming false = 0 and true = 1):

$$1 - \sum_{\sigma \in \bar{L}} \max \left(L(\sigma) - D_\sigma \prod_{\sigma' \cdot \langle t \rangle \cdot \sigma'' = \sigma} \frac{W_t}{\sum_{\left(\boldsymbol{M_0} + \sum_{t'' \in \sigma'} (\boldsymbol{t''^\bullet} - {}^\bullet\boldsymbol{t''})\right) \succeq {}^\bullet\boldsymbol{t'}} W_{t'}}, 0 \right) \tag{4}$$

Then, the full problem can be written as maximising (4) such that (1) $\wedge$ (2) $\wedge$ (3). By construction, the optimal solution to this problem corresponds to a SPN that maximises uEMSC, satisfying the problem of Definition 8.

Irrelevant places. Next to an event log L, the optimisation problem requires a parameter of the maximum number of places the optimiser can utilise. Even if we do not consider arc multiplicities, there are $2^{2|T|}$ potential places in an SPN with transitions T. For 10 transitions, this would already yield 1 048 576 candidate places. The optimisation problem uses $2|T|+1$ variables for each place, thus for practical computabilty, the number of places needs to be limited.

Observe that if an optimal SPN has $|P|$ places, then allowing for more places in will not decrease uEMSC: the optimisation problem can, for instance, simply duplicate places, which obviously does not change the behaviour of the resulting net. We refer to such places as *irrelevant* places:

Definition 9 (Irrelevant place). *Let L be a log and let M, M' be SPNs with places P and P', such that $P = P' \setminus \{p\}$ and* uEMSC(L, M) = uEMSC(L, M'). *Then, p is an* irrelevant *place.*

To check whether a place in an SPN is irrelevant, we can remove this place and compare the uEMSC of the SPN before and after the place removal. We conjecture that if the optimiser returns a net with irrelevant places, then adding more places will not improve uEMSC, and consequently Definition 8 is satisfied.

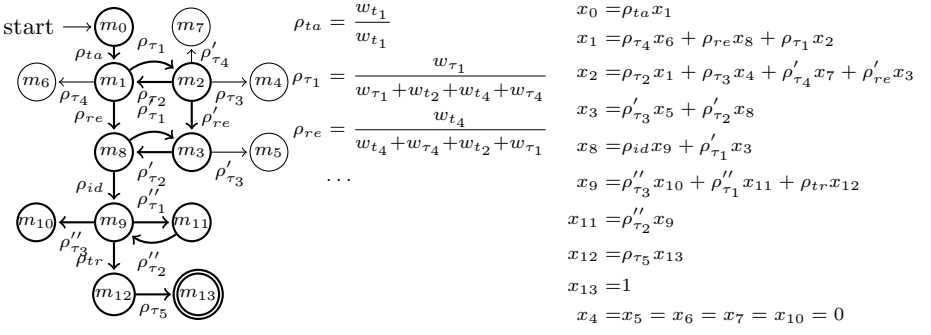
Conjecture 1. If the optimiser returns a irrelevant place, then adding even more places $|P|$ will not increase the uEMSC score.

We use these two results to guide the optimisation: we simply attempt with a number of places $|P|$ and check for irrelevant places. If no such place is found, we repeat the optimisation with a larger number of places, until an optimisation yields an irrelevant place. Then, we have a lower bound and an upper bound for the number of places $|P|$, and we can apply a binary search to find the model with the smallest number of places that satisfies Definition 8. While to the best of our knowledge this is the first stochastic discovery technique that guarantees an as-high-as-possible uEMSC score, it requires a mixed-integer non-convex solver and would probably need further consideration to be practically applicable. Therefore, we leave its implementation as future work. Instead, we proceed with a more practically applicable technique that works on the more generic representational bias of *labelled* Petri nets, and that uses a separately discovered control flow model as input.

6 Stochastic Discovery Given the Control-Flow Structure

In this section, we address a different setting: we assume that a control flow model is given, and we need to assign the stochastic perspective to it that gives us the highest uEMSC. As a trade-off for being given the control flow model, the method of this section can handle labelled transitions. Given a log and a labelled Petri net N, we discover an SLPN that best represents the stochastic information in L with the control flow of N. Our strategy is to turn the net into an SLPN M that assigns a *weight parameter* to every transition. Then, stochastic discovery is posed as an optimisation problem, where values for the weights must be found so that a stochastic conformance measure is maximised.

To set up the optimisation problem, for every trace σ in the log, we extract a symbolic formula characterising the parametric probability of σ according to M, by adapting the trace probability calculation given in [10,11]. First, σ is turned into an SDFA to account for silent transitions. Then, the cross product of the SDFA and a parametric stochastic reachability graph of M is constructed. The cross product is a parametric absorbing Markov chain, from which a system of equations accounting for step-wise probabilities can be extracted, in turn allowing to obtain the probability of σ according to M as the absorption probability of the Markov chain - which can be solved symbolically, considering the parameters. This is then repeated for every trace, combining all the so-obtained symbolic formulae together using a stochastic conformance measure. We detail these steps in the remainder of the section.



(a) The cross-product system. (b) The equation system.

Fig. 5: Example for the trace ⟨talk, record, identify, troubleshoot⟩.

Note that we cannot equate the symbolic formulae to the observed trace probabilities in the log, as due to the representational bias of the given control flow, that solution may not exist.

Constructing the cross-product system. We convert the given net N into a parametric SLPN M by associating each transition in N to a weight parameter. To compare M with the input log L, we characterise the probability of each trace σ in L according to M. This cannot be done directly: if the input N contains silent transitions, the same trace might correspond to infinitely many different paths in M [10,11]. We therefore follow the steps of [10,11], with as main difference that while in [10,11] the weights of the SLPN are given, and the only unknown variable is the trace probability, here also the weights are parameters. So, instead of getting a solution for the trace probability, we will obtain a symbolic formula describing how the trace probability relates to the weight parameters of M.

The first step towards this is to turn each trace σ into an SDFA (cf. Definition 3), then computing the cross-product system of such an SDFA and the stochastic reachability graph (cf. Definition 4) of M, recalled here:

Definition 10 (Cross-Product System). *Let $R = (\Sigma^1, S^1, s_0^1, S_f^1, \varrho, p)$ be the stochastic reachability graph of an SLPN M, and $D(\sigma) = (\Sigma^2, S^2, s_0^2, S_f^2, \delta)$ be an SDFA describing all and only those runs whose corresponding trace is σ. The* cross-product system *of M and σ is an absorbing Markov chain $\mathcal{E}_M^\sigma = M \otimes \sigma = (s_0^\otimes, \varrho^\otimes, p^\otimes, S^\otimes, S_f^\otimes)$ where:*

- $s_0^\otimes = (s_0^1, s_0^2)$,
- *for every $s = (s^1, s^2) \in S^\otimes$, we have $s^1 \in S^1 \wedge s^2 \in S^2$,*
- *for every $s = (s^1, s^2)$ and $s' = (s^{1'}, s^{2'}) \in S^\otimes$ with $\exists_{l \in \Sigma^1 \cap \Sigma^2} \langle s^1, l, s^{1'} \rangle \in \varrho \wedge \delta(s^2, l) = s^{2'}$, we have $\langle s, l, s' \rangle \in \varrho^\otimes$, $p^\otimes(s, l, s') = p(s^1, l, s^{1'})$,*
- *for every $(s_f^1, s_f^2) \in S_f^\otimes$, we have $s_f^1 \in S_f^1 \wedge s_f^2 \in S_f^2$,*

For example, Figure 5a shows the cross-product system of the trace ⟨*talk, record, identify, troubleshoot*⟩ and the stochastic reachability graph in Figure 1b.

It has multiple paths whose label sequence corresponds to the trace, and we omit the parts of the system that do not lead to the accepting state m_{13}.

Describing trace probabilities. Given the cross-product system $\mathcal{E}^{\sigma}_{M} = (s^{\otimes}_{0}, \varrho^{\otimes}, p^{\otimes}, S^{\otimes}, S^{\otimes}_{f})$, we denote $S^{\otimes}_{n}$ as the set of non-target accepting states. To describe how the probability of trace σ according to M depends on the parameters of M, we recast [10,11], which uses standard techniques from literature on absorbing Markov chain to turn the cross-product into a corresponding system $\mathcal{E}^{\sigma}_{M}$ of step-wise equations, where every state $s \in S^{\otimes}$ corresponds to a probability variable x_s, and equations are defined based on $M \otimes \sigma$ as follows:

$$
\begin{aligned}
&x_{s_i} = 1 && \text{for each } s_i \in S^{\otimes}_{f} \setminus S^{\otimes}_{n} \\
&x_{s_j} = 0 && \text{for each } s_j \in S^{\otimes}_{n} \\
&x_{s_k} = \textstyle\sum_{(s_k, s'_k) \in \varrho} p(s_k, s'_k) \cdot x_{s'_k} && \text{for each } s_k \in S^{\otimes} \setminus S^{\otimes}_{f}
\end{aligned}
$$

Recall that the state probability variables and the parameters of M are unknown. In addition, in $\mathcal{E}^{\sigma}_{M}$, variable $x_{s^{\otimes}_{0}}$ denotes the probability of σ according to M. We can then solve the system symbolically, obtaining a formula that relates $x_{s^{\otimes}_{0}}$ to (only) the weight parameters from M. We denote this formula with $\widetilde{p}_M(\sigma)$.

In our running example, the system of equations for the cross-product system of Figure 5a is given in Figure 5b. By symbolically solving the equation system, we derive $(\rho_{re} + \rho_{\tau_1} \cdot \rho'_{re} \cdot \rho'_{\tau_2}) \cdot \rho_{ta} \cdot \rho_{id} \cdot \rho_{tr} \cdot \rho_{\tau_5} / ((1 - \rho_{\tau_1} \cdot \rho_{\tau_2}) \cdot (1 - \rho'_{\tau_1} \cdot \rho'_{\tau_2}) \cdot (1 - \rho''_{\tau_1} \cdot \rho''_{\tau_2}))$, where each ρ is a division of weight parameters; e.g. $\rho_{re} = w_{t_4} / (w_{t_4} + w_{\tau_4} + w_{t_2} + w_{\tau_1})$.

Estimating weights through optimisation. We now combine into a single system all the symbolic probability descriptions derived, as shown before, for every trace in the input log L. For every trace σ in L, we compute the trace probability of σ according to the relative frequency of σ in L (denoted $L(\sigma)$) and compare it with the corresponding symbolic formula $\widetilde{p}_M(\sigma)$. We do so by imposing, overall, optimisation against a stochastic conformance measure applied to M and L, such as uEMSC or ER^{-1}, which is a substitution of the trace probability of $M(\sigma)$ by our symbolic formula $\widetilde{p}_M(\sigma)$. For uEMSC, we get:

$$
\text{maximise } 1 - \sum_{\sigma \in \bar{L}} \max\left(L(\sigma) - \widetilde{p}_M(\sigma), 0\right) \tag{5}
$$

This problem is non-convex due to the many divisions. However, the max can be rewritten into linear constraints, and all parameters are numeric, which makes it suitable for standard solvers. A solution to this problem satisfies Definition 8, when taking the representational bias of SLPNs and uEMSC. Another optimal solution for Definition 8 can be obtained by maximising ER^{-1} in a similar way.

7 Evaluation

The two-stage SLPN discovery approaches were implemented in the ProM framework; their source code and the experiments' scripts are publicly available[6]. A

[6] `https://github.com/promworkbench/SLPNMiner`

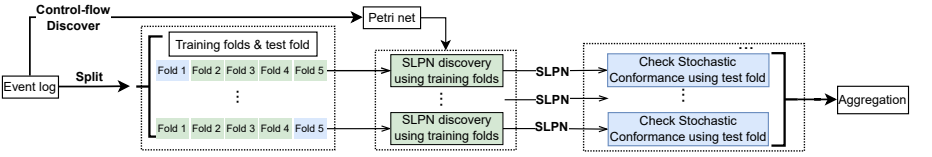


Fig. 6: Experimental setup for evaluation.

Table 1: Results for DFMM [13] control-flow models.

		Stochastic discovery technique						
Log	Measure	d-uEMSC	d-ER	Frequency	Alignment	LH	RH	Scaled
BPIC2013_close	uEMSC	**0.5745**	0.5358	0.0000	0.4273	0.0000	0.0000	0.0000
	EMSC	**0.9059**	0.8820	0.4576	0.4701	0.4600	0.4712	0.4716
	ER^{-1}	0.0881	**0.0906**	0.0397	0.0751	0.0650	0.0650	0.0650
BPIC2013_open	uEMSC	0.3659	0.3910	0.0000	**0.4164**	0.0000	0.0000	0.0000
	EMSC	**0.4897**	0.4541	0.4482	0.4207	0.4209	0.4156	0.4212
	ER^{-1}	0.1082	0.1059	0.0449	**0.1141**	0.1000	0.1000	0.1000
BPIC2017_application	uEMSC	**0.5832**	0.5072	0.2785	0.4117	0.3214	0.3214	0.3214
	EMSC	0.6804	**0.8673**	0.8605	0.8672	0.8672	0.8672	0.8672
	ER^{-1}	0.1134	0.1106	0.1110	**0.1137**	0.1110	0.1110	0.1110
BPIC2017_offer	uEMSC	**0.6563**	0.5828	0.5388	0.5811	0.5373	0.5373	0.5373
	EMSC	**0.9155**	0.9101	0.9026	0.9102	0.9010	0.9010	0.9010
	ER^{-1}	0.2312	0.2330	0.1091	**0.3115**	0.1092	0.1092	0.1092
BPIC2020_domestic	uEMSC	**0.8079**	0.8064	0.0000	0.3575	0.0000	0.0000	0.0000
	EMSC	**0.9158**	0.8596	timeout	timeout	timeout	timeout	timeout
	ER^{-1}	0.1543	**0.1569**	0.0428	0.1144	0.0384	0.0384	0.0384
BPIC2020_request	uEMSC	**0.7537**	0.7151	0.0000	0.6256	0.0000	0.0000	0.0000
	EMSC	0.2830	0.1978	**0.4116**	0.4026	0.3991	0.3990	0.4006
	ER^{-1}	**0.1610**	0.1538	0.0386	0.1220	0.0389	0.0389	0.0389
road traffic fines	uEMSC	**0.8196**	0.3048	0.0139	0.2940	0.0000	0.0000	0.0000
	EMSC	**0.9061**	0.7054	0.5159	0.5311	0.5311	0.5566	0.5403
	ER^{-1}	0.1938	**0.1955**	0.0627	0.1675	0.0590	0.0590	0.0590

60-second timeout is applied to each trace probability calculation. In this section, we compare the quality of models of our techniques with existing stochastic discovery techniques on 7 publicly available real-life event logs[7].

Set-up. Figure 6 shows the experiment setup. Firstly, two control flow discovery algorithms (IMf [9] and DFMM [13]; chosen as they guarantee livelock-freedom) are applied to each full log to obtain a control-flow model. Next, each log is randomly split using 5-fold cross-validation to measure how well the techniques can represent a non-changing process; stochastic discovery (our "d-uEMSC" and "d-ER" discovery techniques, as well as 5 estimators from [6]) are applied to 4 folds and the remaining fold is used to evaluate the SLPN using uEMSC [14], EMSC [14] and ER^{-1} [18]. A 100-second timeout was applied to each conformance measure. The evaluation was repeated 3 times to eliminate random effects: each reported number is thus the average over 15 models.

Results. The results are shown in Tables 1 and 2. We could not compute EMSC on several estimators for the BPIC2020-domestic declations log. Even though d-uEMSC and d-ER optimise for uEMSC and ER^{-1} respectively, training data was not used for measuring, thus in this experiment they are not guaranteed to yield the highest scores. Nevertheless, d-uEMSC got the highest uEMSC value in 9 cases and d-ER got the highest ER^{-1} in 4 cases. On the EMSC measure, for which the techniques did not optimise, d-uEMSC was highest in 9 cases and d-ER in 3 cases. The closest existing technique was the alignment-based estimator, which got a highest EMSC score once, which highlights the trade-offs that need to be made in stochastic discovery, even when optimising for a single measure.

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

Table 2: Results for IMf [9] control-flow models.

		Stochastic discovery technique						
Log	Measure	d-uEMSC	d-ER	Frequency	Alignment	LH	RH	Scaled
BPIC2013_close	uEMSC	**0.4989**	0.3860	0.0000	0.4273	0.0000	0.0000	0.0000
	EMSC	**0.7199**	0.7140	0.4568	0.6569	0.4681	0.4615	0.4609
	ER^{-1}	**0.0856**	0.0835	0.0397	0.0751	0.0650	0.0650	0.0650
BPIC2013_open	uEMSC	0.4391	**0.4518**	0.0000	0.4165	0.0000	0.0000	0.0000
	EMSC	**0.7041**	0.7005	0.4208	0.4483	0.4482	0.4465	0.4385
	ER^{-1}	0.1396	**0.1475**	0.0449	0.1141	0.0999	0.0999	0.0999
BPIC2017_application	uEMSC	0.3256	**0.3753**	0.2785	0.4117	0.3214	0.3214	0.3214
	EMSC	0.8068	**0.8917**	0.8650	0.8663	0.8649	0.8650	0.8590
	ER^{-1}	0.1431	0.1574	0.0609	**0.1686**	0.0457	0.0457	0.0457
BPIC2017_offer	uEMSC	**0.6564**	0.5829	0.5388	0.5811	0.5373	0.5373	0.5373
	EMSC	**0.9155**	0.9101	0.9026	0.9102	0.9010	0.9010	0.9010
	ER^{-1}	0.1080	0.1042	0.1091	0.1037	**0.1092**	**0.1092**	**0.1092**
BPIC2020_domestic	uEMSC	0.0231	0.0152	0.0001	**0.5799**	0.0000	0.0000	0.0000
	EMSC	0.6187	0.5940	0.4342	**0.9361**	timeout	timeout	timeout
	ER^{-1}	0.0763	0.0791	**0.1472**	0.1277	0.0384	0.0384	0.0384
BPIC2020_request	uEMSC	0.2830	0.1978	0.0000	**0.6256**	0.0000	0.0000	0.0000
	EMSC	**0.7537**	0.7151	0.4020	0.3987	0.4093	0.4092	0.3991
	ER^{-1}	0.0701	0.0637	0.0386	**0.1220**	0.0386	0.0386	0.0386
road traffic fines	uEMSC	**0.1729**	0.1407	0.0140	0.0700	0.0000	0.0000	0.0000
	EMSC	0.6148	**0.7473**	0.5567	0.5291	0.5292	0.5310	0.5389
	ER^{-1}	0.1188	0.1389	0.0627	**0.1493**	0.0590	0.0590	0.0590

When comparing the control-flow discovery techniques, DFMM combines well with our d-uEMSC and d-ER, as a highest measure (of all stochastic discovery techniques) is achieved in 16 out of 21 log-measure combinations, while for IMf this is 13. We manually inspected the results for the BPIC2020_domestic log on IMf, where the alignment-based estimator fared better than d-uEMSC and d-ER. We found that the many silent transitions in the model led to a large state space for the cross-product system, which made the derivation of a symbolic representation of trace probability time out. Consequently, these trace probabilities were not considered during optimisation. We also re-ran some instances multiple times, and found that our solver may return different values over different runs and, as expected, does not guarantee optimality on our non-convex problem.

In summary, the estimators proposed in this paper can be applied to real-life logs and discover better SLPNs, considering common stochastic conformance measures, even the ones they did not optimise for. Thus, they provide alternative estimation approaches to existing two-stage stochastic discovery techniques.

8 Conclusion

In this paper, we formally defined stochastic process discovery as finding a model with an optimal conformance checking measure over a given representation bias. We studied the implications of this definition in detail, and introduced techniques for two biases: one for SPNs (a one-stage approach), and one that takes a control-flow model for SLPNs (a two-stage approach). We implemented and evaluated the latter one for uEMSC and ER^{-1}, and found that they perform well compared to existing techniques, even evaluating stochastic measures on test logs they did not optimise for. A direction for future work is to improve the implementation of symbolic trace probabilities, and to implement the one-stage approach. Furthermore, stochastic discovery can be extended to optimisation problems over further conformance measures, and on combinations of such measures. In particular, considering simplicity measures, which must be invented first for stochastic models, may prove beneficial for stochastic process discovery.

References

1. van der Aalst, W.M.P.: Process Mining - Data Science in Action, Second Edition. Springer (2016)
2. Alkhammash, H., Polyvyanyy, A., Moffat, A., García-Bañuelos, L.: Entropic relevance: A mechanism for measuring stochastic process models discovered from event data. Inf. Syst. **107** (2022)
3. Bause, F., Kritzinger, P.S.: Stochastic Petri nets - an introduction to the theory (2. ed.). Vieweg (2002)
4. Bergami, G., Maggi, F.M., Montali, M., Peñaloza, R.: Probabilistic trace alignment. In: ICPM. IEEE (2021)
5. Brockhoff, T., Uysal, M.S., van der Aalst, W.M.P.: Time-aware concept drift detection using the earth mover's distance. In: ICPM. IEEE (2020)
6. Burke, A., Leemans, S.J.J., Wynn, M.T.: Stochastic process discovery by weight estimation. In: ICPM Workshops. LNBIP, vol. 406. 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. Springer (2021)
8. Jansen, N., Junges, S., Katoen, J.: Parameter synthesis in markov models: A gentle survey. In: Principles of Systems Design - Essays Dedicated to Thomas A. Henzinger on the Occasion of His 60th Birthday. LNCS, vol. 13660. Springer (2022)
9. Leemans, S.J.J., Fahland, D., van der Aalst, W.M.P.: Discovering block-structured process models from incomplete event logs. In: Petri Nets. LNCS, vol. 8489. Springer (2014)
10. Leemans, S.J.J., Maggi, F.M., Montali, M.: Reasoning on labelled Petri nets and their dynamics in a stochastic setting. In: BPM. LNCS, vol. 13420. Springer (2022)
11. Leemans, S.J.J., Maggi, F.M., Montali, M.: Enjoy the silence: Analysis of stochastic Petri nets with silent transitions. CoRR **abs/2306.06376** (2023)
12. Leemans, S.J.J., Mannel, L.L., Sidorova, N.: Significant stochastic dependencies in process models. Inf. Syst. **118** (2023)
13. Leemans, S.J.J., Poppe, E., Wynn, M.T.: Directly follows-based process mining: Exploration & a case study. In: ICPM. IEEE (2019)
14. Leemans, S.J.J., Syring, A.F., van der Aalst, W.M.P.: Earth movers' stochastic conformance checking. In: BPM Forum. LNBIP, vol. 360. Springer (2019)
15. Marsan, M.A., Conte, G., Balbo, G.: A class of generalized stochastic Petri nets for the performance evaluation of multiprocessor systems. ACM Trans. Comput. Syst. **2**(2) (1984)
16. Mazak, A., Wolny, S., Wimmer, M.: On the need for data-based model-driven engineering. In: Security and Quality in Cyber-Physical Systems Engineering, pp. 103–127. Springer (2019)
17. Molloy, M.K.: Performance analysis using stochastic Petri nets. IEEE Trans. Computers **31**(9) (1982)
18. Polyvyanyy, A., Moffat, A., García-Bañuelos, L.: An entropic relevance measure for stochastic conformance checking in process mining. In: ICPM. IEEE (2020)
19. Rogge-Solti, A., van der Aalst, W.M.P., Weske, M.: Discovering stochastic Petri nets with arbitrary delay distributions from event logs. In: BPM workshops. LNBIP, vol. 171. Springer (2013)
20. Tax, N., Lu, X., Sidorova, N., Fahland, D., van der Aalst, W.M.P.: The imprecisions of precision measures in process mining. Inf. Process. Lett. **135** (2018)
21. van der Werf, J.M.E.M., van Dongen, B.F., Hurkens, C.A.J., Serebrenik, A.: Process discovery using integer linear programming. Fundam. Inform. **94**(3-4) (2009)