Process Discovery as a Global Place Combination Problem

Tobias Brockhoff, Christopher T. Schwanen, Lisa L. Mannel, and Wil M. P. van der Aalst

Chair of Process and Data Science (PADS), RWTH Aachen University, Aachen, Germany

Abstract

Petri net synthesis aims to construct a precise and fitting Petri net from a given specification, but typically requires the latter to be complete and free of noise. This requirement hampers its application to process discovery from real-world data which are often incomplete and noisy. Methods such as the eST-Miner leverage synthesis ideas allowing places which do not fit the data up to a threshold. However, composing individually fitting places into a Petri net can result in models with global behavioral problems such as deadlocks. This paper addresses this composition problem by proposing a new discovery method that, given a set of place candidates, aims to find the subset of places such that the resulting Petri net maximizes existing conformance checking scores. Specifically, the proposed approach uses Bayesian optimization to directly optimize such measures, yet does not guarantee optimal solutions. Our evaluation shows that, despite being computationally expensive, this method can successfully fine-tune Petri nets discovered by the eST-Miner with respect to state-of-the-art conformance checking measures.

Keywords

Process Mining, Process Discovery, Petri Nets, Workflow Nets, Bayesian Optimization

1. "The whole is greater than the sum of its parts"

Unfortunately, this (mis)quote of Aristotle does not necessarily apply when discovering Petri nets from event data by synthesizing fitting places and composing them into a Petri net. The goal of Petri net synthesis [8] is to construct a Petri net that can model the entire behavior in a given specification while introducing as little additional behavior as possible.

In process discovery [1, 3, 6], we aim to find a process model that represents a given event log in the most suitable way. The extent to which a process model is considered suitable usually depends on some predefined metric based on one or a combination of the four model quality dimensions [1]: (i) fitness, (ii) precision, (iii) simplicity, and (iv) generalization. Throughout the years, a variety of quality metrics have been identified for the different dimensions, with fitness usually being the main driver in process discovery [13]. This is also reflected in the number of metrics available: there are many metrics for fitness and precision, but noticeably fewer for simplicity and generalization. The way different metrics are determined based on various techniques such as token-based replay [28] or alignments [4] to compute fitness and precision, or structural analyses of the model to compute simplicity [17]. Nonetheless, all of the different quality metrics have certain drawbacks [29, 31, 32] and their selection often depends on the chosen process discovery technique.

Generally, starting with an event log, we aim for a process model optimized over some (combined) quality metric that evaluates the discovered model based on the event log. Since an event log can be considered a language-based specification, we can use language-based region theory [10, 20] to synthesize a Petri net that best describes the specified behavior. In such a setting, the set of transitions in the Petri net is typically already fixed, but we are looking to connect them to specific places to restrict the behavior allowed by the model in a way that serves the objective. The ILP Miner [35, 36], for example, uses *Integer Linear Programming* (ILP) to iteratively add new places corresponding to a minimal region in the event log language, but is strongly influenced by the places discovered first.

Algorithms & Theories for the Analysis of Event Data 2026

brockhoff@pads.rwth-aachen.de (T. Brockhoff); schwanen@pads.rwth-aachen.de (C. T. Schwanen); mannel@pads.rwth-aachen.de (L. L. Mannel); wvdaalst@pads.rwth-aachen.de (W. M. P. van der Aalst)

0000-0002-6593-9444 (T. Brockhoff); 0000-0002-3215-7251 (C. T. Schwanen); 0000-0002-0955-6940 (W. M. P. van der Aalst)

In general, such approaches often struggle with noise in real-world event logs and the exponential growth of different candidate places. This renders these approaches far from trivial and forces us to employ heuristics or exploit monotonicity properties to traverse the search space more efficiently [2]. This idea is picked up by the eST-Miner [21] which evaluates places individually and composes them into a Petri net. Whenever a candidate is excluded, it also prunes the search tree. This local perspective might cause structural and semantic problems such as deadlocks, low fitness or unsoundness in the naively composed model. In [22, 23], the authors address yet do not fully remedy these problems. Besides, the resulting net may still reflect a non-deterministically selected local optimum rather than the best possible solution.

In this work, we propose an optimization-based process discovery approach inspired by the ILP Miner and eST-Miner. In contrast to these miners, which evaluate individual places with respect to the given event log, our approach evaluates possible combinations of a given set of candidate places globally to return the best model based on the given configuration and set of input places. While it cannot compete with traditional approaches in terms of computational efficiency, it allows for a very flexible selection and prioritization of desirable model properties. Thus, the resulting models it produces might also serve as a baseline when evaluating model quality in a large variety of application contexts.

2. Blackbox Optimization of Complex Objective Functions

Common fitness and precision measures do not directly depend on the structure of the Petri net but rather on its language, leading to complex relationships between quality scores and the structure of general Petri nets. Evaluating quality metrics locally (e.g., only for a single place) might poorly approximate the quality scores commonly employed in process mining (e.g., alignment- or token-based fitness and precision).

To address this challenge, we investigate whether we can directly optimize existing quality metrics (without guaranteed optimality). Given that alignment- and token-based replay metrics lack additional information such as tractable derivatives with respect to the places, we turn to *Black-Box Optimization* (BBOpt). BBOpt algorithms—such as simulated annealing, evolutionary algorithms, random search, and *Bayesian optimization* (BO)—optimize an objective function based solely on evaluations at selected points. In our case, they would select candidate places for evaluation of the resulting Petri net's score.

Key considerations for implementing a BBOpt approach for global place discovery include: (i) high dimensionality (i.e., potentially over 20 places) (ii) binary decision variables (one per place), and (iii) costly objective function evaluations. Specifically, evaluating alignments or conducting token-based replay is computationally expensive compared to simpler evaluations found in combinatorial optimization (e.g., cost of a route in traveling salesperson problems) or games (e.g., determining if a game state is winning or losing). This motivates our use of BO methods known for their sample efficiency when optimizing expensive blackbox functions [30]. However, classic BO typically struggles with high-dimensional discrete spaces [12, 14]—a challenge recently addressed in the machine learning community [14, 26, 34].

We assume the same setting as the eST-Miner, that is, we synthesize a Petri net (P, T, F) where the set of transitions is already fixed (one transition per activity in the event log L plus artificial start and end activities), and we aim for the best set of places P. However, instead of considering all potential places (i.e., all combinations of input and output transitions), we employ a place oracle to provide a candidate set P_{cand} of promising places from which the best set of places $P \subseteq P_{cand}$ is chosen.

Our implementation spans three main components: the place oracle, the BBOpt procedure over sets of places, and a Petri net evaluation module. As place oracle, we employ SPECpp[1], a reference implementation of different eST-Miner variants. Besides discovering a Petri net, SPECpp can export the places it has identified. We use this exported place specification or combinations of the former as input for the BBOpt component. For the BBOpt, we formulate our place composition task as an additional experiment into the implementation[2] of [14]. In particular, we use a linear combination of the Petri

[1] https://github.com/promworkbench/SPECpp
[2] https://github.com/facebookresearch/bo_pr

net's F-score and simplicity measure as objective function:

$$\underset{P \subseteq P_{cand}}{\arg\max} \; (1-\lambda) \cdot \text{F-score}((P, T_P, F_P), L) + \lambda \cdot \text{simplicity}((P, T_P, F_P), L), \tag{1}$$

where $0 \leq \lambda \leq 1$, and T_P and F_P are the resulting transitions and arcs with respect to the chosen set of places P. Here, $T_P \subseteq T$ is the subset of transitions that are adjacent to at least one place. While the F-score is considered in most works on process discovery to describe a process model's quality by means of a single value, initial experiments indicate that incorporating simplicity acts as a regularizer on the Petri net's structure, for example, aiding in removing implicit places. Candidate models are internally evaluated using ProM. In particular, we use the alignment-based fitness implementation from [15] and ETC-precision [5] as conformance measures. Moreover, we adopt PM4Py's default *mean node degree* simplicity measure which is computed as $1/\,(1 + \max\{\text{mean node degree} - 2, 0\})$ [11], a reciprocal penalty function based on how severely the mean node degree deviates from the simplest non-trivial place/transition with one input and one output arc.

A further important aspect of the implementation is the removal of disconnected transitions. While unconstrained transitions may be meaningful in some settings, they typically deteriorate precision in practical applications. Therefore, SPECpp removes such transitions by default, and we adopt the same behavior. This design choice also creates an interesting degree of freedom for BBOpt: by deliberately leaving specific transitions disconnected during the optimization, the method can effectively decide to remove transitions from the model. One the one hand, since we compute a model's quality based on the original event log, this translates into log moves for the left out activities. On the other hand, this can be very beneficial if, given the respective place candidates, it is impossible to otherwise sufficiently constrain a transition without generating too many escaping edges in the ETC-precision computation.

3. Evaluation

Our evaluation is founded on two main research questions:

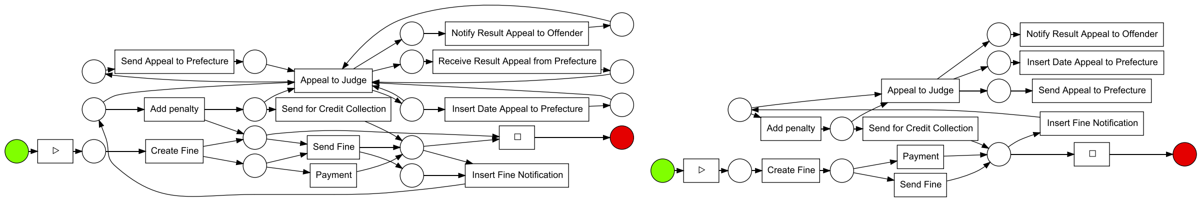
1. Can we compose high-quality Petri nets from good-quality place candidates using BBOpt?
2. How do the Petri nets synthesized using eST-Miner and BBOpt compare to models discovered by state-of-the-art process discovery approaches?

We focus on good-quality place candidates for three reasons. First, we applied two BBOpt methods under a time budget of 30 minutes per place set, and optimizing over hundreds of sets would be prohibitive. Second, prior work [14] indicates that contemporary BO techniques can handle problems with a few dozen binary variables. Substantially larger place sets likely exceed their practical capabilities. Third and most importantly, we consider it more insightful to search for non-trivial solutions—that is, models that do not include *all* available places—that outperform the best models produced by the eST-Miner. This way we hope to push the boundaries of Petri net synthesis for process discovery.

For comparison, we consider Fodina [33], *Split Miner* (SM) [7] version 1 and 2, Prime Miner [9], *Inductive Miner–infrequent* (IMf) [18], and several eST-Miner variants [16, 21–23]. As input data, we use four event logs: *Road Traffic Fine Management Process* (RTFM) [19], Sepsis [25], Teleclaims [1], and Repair Example[3]. In contrast to the latter two logs, RTFM and Sepsis are at least based on real-world process data. These logs were selected because they can be processed by most discovery algorithms without substantial additional preprocessing, while still yielding process models of decent quality that are suitable for a meaningful quality metric evaluation. Because not all considered discovery algorithms support lifecycle events, we filter the artificial logs so that they contain only complete lifecycle events.

For the eST-Miner family of methods, we systematically vary the noise threshold $\tau \in \{0.8, 0.9, 1\}$, the maximum tree depth $\{3, 4, 5, 6, \infty\}$, and we place fitness criterion, *absolute* or *aggregated* place fitness criteria as proposed in [23]. For the δ-variant, we employ a linear delta function with gradient 1 and $\delta = 0.2$, as this configuration has shown robust performance in [23]. For the ETC-variant [16], we consider

[3]https://promtools.org/prom-6-tutorial/



(a) Best eST-Miner Petri net

(b) Best BOPR Petri net

Figure 1: Best models discovered by (a) eST-Miner and (b) BOPR on the top-3 places set for the RTFM event log.

$\rho \in \{0.9, 0.98\}$. In all eST-Miner configurations except the ETC-variant, we enable concurrent replay-based implicit place removal [24], which prunes the candidate place set on-the-fly while discovering new places. This mechanism is crucial to keep the number of discovered places manageable, so that the resulting place sets can be handled not only by the BBOpt methods but also by SPECpp's default post-processing pipeline, including ILP-based implicit place removal. Empirically, concurrent implicit place removal can reduce the number of discovered places by orders of magnitude. The ETC-variant uses an alternative strategy to avoid implicit places and typically produces reasonably small place sets on its own. For all place discovery runs, we impose a timeout of 2 minutes.

For the baseline discovery algorithms, we tune parameters with respect to the objective function in Equation (1) in a way that aligns with prior work. For SM version 1, we explore $\epsilon \in \{0.01, 0.05, 0.1, 0.15, 0.2\}$ and $\eta \in \{0.2, 0.3, 0.4, 0.5, 0.6\}$, thereby exhaustively testing configurations around and including the optimal setting reported in [7]. For version 2, we tested $\epsilon \in \{0.2, 0.3, \dots, 0.9\}$. For IMf, we vary the $f \in \{0.05, 0.1, 0.2, 0.3, 0.4\}$. For Prime Miner, we rely on the corresponding ProM plugin that returns a single model for each event log. In contrast to SM and IMf, Fodina exposes a large number of hyperparameters; we therefore use the provided defaults as an exhaustive parameter study would be beyond the scope of this work.

For our approach we consider three systematically constructed sets of places based on SPECpp. First, we consider the places associated with the best eST-Miner model for the respective log before SPECpp's post-processing (including particularly ILP-based implicit place removal). Second, we construct a merged place set obtained by combining the places from the three top-performing eST-Miners. Third, in order to stress-test the BBOpt step on relatively large place sets, we form a union of all place sets discovered with a maximum tree depth of 3. We tested two BBOpt methods: *Bayesian optimization via probabilistic reparameterization* (BOPR) [14] and an evolutionary algorithm based on the nevergrad library [27]. Each method directly optimized the objective function given in Equation (1) for $\lambda = 0.2$. Each method had access to 32 Intel Xeon Platinum 8468 processing cores. Moreover, BOPR, which supports GPU usage, was given access to a single Nvidia L40S GPU. Finally, each BBOpt experiment was repeated 5 times for different random seeds.

In total, we evaluated 762 Petri nets. To aggregate these experimental results, we first removed all runs for which we could not compute a valid F-score. Empirically, such invalid scores arose when the terminal language of the discovered model was empty—that is, the model is not relaxed sound. Consequently, we discarded Prime Miner and Fodina for Sepsis. In a second step, for eST-Miner, SM, Fodina, IMf, we selected, for each event log, the best model according to the aggregated quality metric defined in Equation (1) with $\lambda = 0.2$ among all tested parameter settings. This model is then reported together with the corresponding configuration. For IMf, the reported configuration is the noise threshold; for SM (v. 1), it is the value of ϵ/η. For eST-Miner, we additionally report the selected variant, the fitness criterion, and the tree depth. For the proposed approach, we report the repetition that achieved the median value of the aggregated quality metric, together with the difference between this median run and the best run observed. Since BOPR consistently performed at least as well as nevergrad, we only report results for BOPR. Table 1 summarizes the resulting scores and highlights, in bold, the best values for each quality measure and event log.

In general, BOPR outperforms the best eST-Miner model on at least the top and top-3 place sets for three of the four event logs and performs equally well on the fourth. These top-k place sets are

Table 1
Quality metrics for BOPR and the best models discovered by the baseline process discovery approaches. We computed the aggregated quality metric (QM) for $\lambda = 0.2$. The best results (up to some small difference) for each metric and event log are highlighted in bold. For BOPR, we show median results as well as the respective difference (Δ-QM) to the best aggregated quality metric obtained. Moreover, we show how many of the respective candidates places where selected in the run achieving median QM score.

Event Log	Algorithm	Config	Fit.	Prec.	Simp.	F-Score	QM	Δ-QM	Place Info
Repair Example	**BOPR**	**best**	**0.99**	0.99	**0.83**	**0.99**	**0.96**	0.01	8/14
		d4	**0.99**	0.99	**0.83**	**0.99**	**0.96**	0	8/24
		top3	**0.99**	0.99	**0.83**	**0.99**	**0.96**	0	8/18
	SMv1	**0.2/0.2**	0.86	**0.99**	**0.83**	0.92	0.9	-	
	SMv2	**0.9**	0.98	0.94	**0.83**	0.96	0.94	-	
	eST-Delta	**Abs-10000**	**0.99**	0.99	**0.83**	**0.99**	**0.96**	-	
	Fodina	-	**0.99**	0.96	0.76	0.97	0.93	-	
	IMf	**0.3**	**0.99**	0.98	0.8	0.99	0.95	-	
RTFM	**BOPR**	**best**	0.88	**1**	0.76	0.93	0.9	0.01	11/29
		d4	0.88	**1**	0.37	0.93	0.82	0	29/61
		top3	0.88	**1**	0.73	0.93	0.89	0.03	12/31
	SMv1	**0.01/0.2**	0.68	0.64	0.73	0.66	0.67	-	
	SMv2	**0.4**	0.88	**1**	**0.77**	**0.94**	**0.9**	-	
	eST-Standard	**Abs-5**	0.88	**1**	0.58	0.93	0.86	-	
	Fodina	-	0.84	0.95	0.65	0.89	0.84	-	
	IMf	**0.3**	**0.93**	0.83	0.71	0.88	0.85	-	
Sepsis	**BOPR**	**best**	0.42	1	**1**	0.59	0.67	0	5/6
		d4	0.69	0.79	0.32	**0.73**	0.65	0.01	31/97
		top3	0.49	0.76	**1**	0.6	0.68	0	6/10
	SMv1	**0.2/0.6**	0.83	0.52	0.67	0.64	0.64	-	
	SMv2	**0.6**	0.66	0.82	0.58	0.73	**0.7**	-	
	eST-ETC	**Agg-3**	0.42	**1**	**1**	0.59	0.67	-	
	IMf	**0.2**	**0.9**	0.53	0.61	0.67	0.66	-	
Teleclaims	**BOPR**	**best**	0.82	0.95	**1**	0.88	0.91	0	8/20
		d4	0.85	**1**	0.43	0.92	0.82	0.03	20/49
		top3	0.85	**1**	0.62	0.92	0.86	0.04	14/26
	SMv1	**0.01/0.2**	0.85	**1**	0.92	0.92	0.92	-	
	SMv2	**0.2**	0.82	**1**	**1**	0.9	0.92	-	
	eST-ETC	**Abs-5**	0.85	**1**	0.55	0.92	0.85	-	
	Fodina	-	**1**	**1**	0.76	**1**	**0.95**	-	
	IMf	**0.3**	0.93	**1**	0.88	0.97	0.95	-	

relatively small; however, the selected places still constitute non-trivial subsets of the available candidate places (comp. last column of Table 1). BOPR also performs strongly on the depth-restricted place set, empirically indicating that it can handle larger place sets of size 50−100 without a substantial loss in quality. Regarding synthesis-based process discovery, SM still achieves the best average performance. Nevertheless, for the RTFM event log, the best model according to the aggregated quality metric is obtained by BOPR on the top-3 place set, with a score of 0.92 (obtained by summing the QM and Δ-QM columns). This model, depicted in Figure 1, is visually simpler than the best eST-Miner model.

Interestingly, this Petri net contains a dead part related to the appeals. Although this does not align with our initial motivation to obtain models without dead parts, in this case the dead part improves PM4Py's default *mean node degree* simplicity. Besides, keeping this part does not affect the F-score because it leaves the behavior unchanged; thus, it actually improves the global quality score. While the result may still correspond to a local optimum, this behavior can be interpreted as a loophole exploited by the optimization, revealing conceptual limitations of existing standard measures.

This observation raises the question of whether the proposed method might be easily adapted to obtain models without dead parts. To investigate this, we conducted preliminary experiments on the place set underlying the model in Figure 1, where we changed the simplicity measure. Due to space limitations, these results are not reported in detail. Because the number of possible places is bounded by the size of the candidate set, one can, for example, use the fraction of included places to measure simplicity. This formulation encourages the optimization algorithm to produce models that are fitting and precise while containing few places. Similarly, we experimented with minimizing the fraction of transitions, all nodes (places and transitions), or arcs included. Initial results indicate that these alternative simplicity measures can reduce the size of dead parts; however, completely removing them proved surprisingly difficult. The best result was obtained by minimizing the number of places only. This approach removed all dead elements except for the "Appeal to Judge" transition. As can be seen in Figure 1, removing this transition requires to delete two arcs thereby jointly replacing two places. Although the required places are contained in the candidate set, the optimization algorithm did not identify this solution. When minimizing only the number of places, such a one-to-one substitution does not change the score. Even when minimizing the total number of nodes or arcs, the transition was never removed and the results were typically arguably worse. This may be explained by the fact that, in both cases, normalization occurs over a significantly larger population, so small changes have less impact on the global score. Consequently, the effective range of the simplicity score may become quite small for reasonably precise and well-fitting models. Moreover, both places must be replaced at the same time, replacing either place significantly affects precision. Overall, these observations highlight that defining metrics that work well jointly and determining how to combine them properly is a non-trivial problem. Finally, note that for this and other practically well-tractable Petri nets encountered in process discovery, a reachability analysis could efficiently detect such dead transitions. This suggests that combining BBOpt with Petri-net-aware techniques is an interesting direction for future research.

4. Discussion and Future Work

Our results indicate that BBOpt methods such as BOPR can discover locally optimal Petri nets from place candidates with respect to the global quality metrics. Since, from a BBOpt point of view, BOPR only "sees" binary variables, optimization over the inclusion of richer process structures becomes an interesting direction for future research. The experiments suggest that finding good combinations of a few dozen process structures might be within practical reach. Nevertheless, even though BOPR improves upon the models obtained by eST-Miner, process discovery algorithms such as SM and IMf often remain more effective in practice. They tend to be faster while frequently achieving comparable or better models, partly due to modeling advantages such as the ability to introduce silent transitions to represent optional behavior. It is, however, noteworthy that although SM itself is computationally efficient, searching its hyperparameter space can become costly. Some discovered models in our experiments were difficult and time-consuming to evaluate using alignment-based conformance measures.

A threat to validity is that the scores obtained for SM, in particular version 1, are lower than expected [cf. 7]. In our experiments, we directly invoked the SM's functionalities instead of using the command-line entry point. While we made a best effort to replicate the original behavior, our implementation and unintentionally employing different flags than those used in [7] could explain the observed discrepancies. A second threat relates to the evaluation of BOPR for process discovery: starting from eST-Miner, we only considered a limited number of relatively small place sets for a small set of event logs. This restriction limits the generalizability of our findings.

Declaration on Generative AI

During the preparation of this work, the authors used RWTHgpt (powered by GPT-4.1-mini, GPT-o4-mini, and GPT-5.3-chat) in order to: paraphrase and reword as well as to improve writing style. After using this service, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] W. M. P. van der Aalst. *Process Mining: Data Science in Action*. 2nd ed. Berlin, Heidelberg: Springer, 2016. DOI: 10.1007/978-3-662-49851-4.

[2] W. M. P. van der Aalst. "Discovering the "Glue" Connecting Activities." In: *It's All About Coordination: Essays to Celebrate the Lifelong Scientific Achievements of Farhad Arbab*. Vol. 10865. LNCS. Cham: Springer International Publishing, 2018, pp. 1–20. DOI: 10.1007/978-3-319-90089-6_1.

[3] W. M. P. van der Aalst. "Foundations of Process Discovery." In: *Process Mining Handbook*. Vol. 448. LNBIP. Cham: Springer International Publishing, 2022, pp. 37–75. DOI: 10.1007/978-3-031-08848-3_2.

[4] W. M. P. van der Aalst, A. Adriansyah, and B. F. van Dongen. "Replaying history on process models for conformance checking and performance analysis." In: *WIREs Data Mining and Knowledge Discovery* 2.2 (2012), pp. 182–192. DOI: 10.1002/widm.1045.

[5] A. Adriansyah, J. Munoz-Gama, J. Carmona, B. F. van Dongen, and W. M. P. van der Aalst. "Alignment Based Precision Checking." In: *Business Process Management Workshops*. BPM 2012 International Workshops. Vol. 132. LNBIP. Berlin, Heidelberg: Springer, 2013, pp. 137–149. DOI: 10.1007/978-3-642-36285-9_15.

[6] A. Augusto, J. Carmona, and E. Verbeek. "Advanced Process Discovery Techniques." In: *Process Mining Handbook*. Vol. 448. LNBIP. Cham: Springer International Publishing, 2022, pp. 76–107. DOI: 10.1007/978-3-031-08848-3_3.

[7] A. Augusto, R. Conforti, M. Dumas, M. La Rosa, and A. Polyvyanyy. "Split miner: automated discovery of accurate and simple business process models from event logs." In: *Knowledge and Information Systems* 59 (2019), pp. 251–284. DOI: 10.1007/s10115-018-1214-x.

[8] E. Badouel, L. Bernardinello, and P. Darondeau. *Petri Net Synthesis*. Texts in Theoretical Computer Science. An EATCS Series. Berlin, Heidelberg: Springer, 2015. DOI: 10.1007/978-3-662-47967-4.

[9] R. Bergenthum. "Prime Miner - Process Discovery using Prime Event Structures." In: *International Conference on Process Mining*. ICPM 2019. IEEE, 2019, pp. 41–48. DOI: 10.1109/ICPM.2019.00017.

[10] R. Bergenthum, J. Desel, R. Lorenz, and S. Mauser. "Process Mining Based on Regions of Languages." In: *Business Process Management*. BPM 2007. Vol. 4714. LNCS. Berlin, Heidelberg: Springer, 2007, pp. 375–383. DOI: 10.1007/978-3-540-75183-0_27.

[11] A. Berti, S. J. van Zelst, and D. Schuster. "PM4Py: A process mining library for Python." In: *Software Impacts* 17 (2023), p. 100556. DOI: 10.1016/j.simpa.2023.100556.

[12] M. Binois and N. Wycoff. "A Survey on High-dimensional Gaussian Process Modeling with Application to Bayesian Optimization." In: *ACM Trans. Evol. Learn. Optim.* 2.2 (Aug. 2022). DOI: 10.1145/3545611.

[13] J. C. A. M. Buijs, B. F. van Dongen, and W. M. P. van der Aalst. "On the Role of Fitness, Precision, Generalization and Simplicity in Process Discovery." In: *On the Move to Meaningful Internet Systems: OTM 2012*. CoopIS 2012. Vol. 7565. LNCS. Berlin, Heidelberg: Springer, 2012, pp. 305–322. DOI: 10.1007/978-3-642-33606-5_19.

[14] S. Daulton, X. Wan, D. Eriksson, M. Balandat, M. A. Osborne, and E. Bakshy. "Bayesian Optimization over Discrete and Mixed Spaces via Probabilistic Reparameterization." In: *Advances in Neural Information Processing Systems*. NeurIPS 2022. Vol. 35. Curran Associates, Inc., 2022, pp. 12760–12774. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/531230cfac80c65017ad0f85d3031edc-Paper-Conference.pdf.

[15] B. F. van Dongen. "Efficiently Computing Alignments. Using the Extended Marking Equation." In: *Business Process Management*. BPM 2018. Vol. 11080. LNCS. Cham: Springer International Publishing, 2018, pp. 197–214. DOI: 10.1007/978-3-319-98648-7_12.

[16] F. C. Groß, L. L. Mannel, and W. M. P. van der Aalst. "Enhancing the Applicability of the eST-Miner: Efficient Precision-Guided Implicit Place Avoidance." In: *Proceedings of the 5th International Conference on Process Mining*. ICPM 2023. IEEE, 2023, pp. 121–128. DOI: 10.1109/ICPM60904.2023.10271973.

[17] K. B. Lassen and W. M. P. van der Aalst. "Complexity metrics for Workflow nets." In: *Information and Software Technology* 51.3 (2009), pp. 610–626. DOI: 10.1016/j.infsof.2008.08.005.

[18] S. J. J. Leemans, D. Fahland, and W. M. P. van der Aalst. "Discovering Block-Structured Process Models from Event Logs Containing Infrequent Behaviour." In: *Business Process Management Workshops*. BPM 2013 International Workshops. Vol. 171. LNBIP. Cham: Springer International Publishing, 2014, pp. 66–78. DOI: 10.1007/978-3-319-06257-0_6.

[19] M. de Leoni and F. Mannhardt. *Road Traffic Fine Management Process*. Eindhoven University of Technology, 2015. DOI: 10.4121/uuid:270fd440-1057-4fb9-89a9-b699b47990f5.

[20] R. Lorenz, S. Mauser, and G. Juhás. "How to synthesize nets from languages - a survey." In: *Winter Simulation Conference*. WSC 2007. IEEE, 2007, pp. 637–647. DOI: 10.1109/WSC.2007.4419657.

[21] L. L. Mannel and W. M. P. van der Aalst. "Finding Complex Process-Structures by Exploiting the Token-Game." In: *Application and Theory of Petri Nets and Concurrency*. Petri Nets 2019. Vol. 11522. LNCS. Cham: Springer International Publishing, 2019, pp. 258–278. DOI: 10.1007/978-3-030-21571-2_15.

[22] L. L. Mannel and W. M. P. van der Aalst. "Discovering Process Models with Long-Term Dependencies While Providing Guarantees and Handling Infrequent Behavior." In: *Application and Theory of Petri Nets and Concurrency*. Petri Nets 2022. Vol. 13288. LNCS. Cham: Springer International Publishing, 2022, pp. 303–324. DOI: 10.1007/978-3-031-06653-5_16.

[23] L. L. Mannel and W. M. P. van der Aalst. "Discovering Process Models with Long-Term Dependencies while Providing Guarantees and Filtering Infrequent Behavior Patterns." In: *Fundamenta Informaticae* 190.2-4 (2024), pp. 109–158. DOI: 10.3233/fi-242168.

[24] L. L. Mannel, R. Bergenthum, and W. M. P. van der Aalst. "Removing Implicit Places Using Regions for Process Discovery." In: *Algorithms & Theories for the Analysis of Event Data 2020*. Vol. 2625. CEUR Workshop Proceedings. Aachen: CEUR-WS.org, 2020, pp. 20–32. URL: https://ceur-ws.org/Vol-2625/paper-02.pdf.

[25] F. Mannhardt. *Sepsis Cases - Event Log*. Eindhoven University of Technology, 2016. DOI: 10.4121/uuid:915d2bfb-7e84-49ad-a286-dc35f063a460.

[26] C. Oh, J. M. Tomczak, E. Gavves, and M. Welling. "Combinatorial Bayesian optimization using the graph cartesian product." In: *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2019.

[27] J. Rapin and O. Teytaud. *Nevergrad - A gradient-free optimization platform*. https://GitHub.com/FacebookResearch/Nevergrad. 2018.

[28] A. Rozinat and W. M. P. van der Aalst. "Conformance checking of processes based on monitoring real behavior." In: *Information Systems* 33.1 (2008), pp. 64–95. DOI: 10.1016/j.is.2007.07.001.

[29] P. Schalk, A. Burke, and R. Lorenz. "Navigating Complexity: Comparing Complexity Measures With Weyuker's Properties." In: *2024 6th International Conference on Process Mining*. ICPM 2024. IEEE, 2024, pp. 145–152. DOI: 10.1109/ICPM63005.2024.10680655.

[30] J. Snoek, H. Larochelle, and R. P. Adams. "Practical Bayesian Optimization of Machine Learning Algorithms." In: *Advances in Neural Information Processing Systems*. Vol. 25. Curran Associates, Inc., 2012. URL: https://proceedings.neurips.cc/paper_files/paper/2012/file/05311655a15b75fab86956663e1819cd-Paper.pdf.

[31] A. F. Syring, N. Tax, and W. M. P. van der Aalst. "Evaluating Conformance Measures in Process Mining Using Conformance Propositions." In: *Transactions on Petri Nets and Other Models of Concurrency XIV*. Vol. 11790. LNCS. Berlin, Heidelberg: Springer, 2019. Chap. 8, pp. 192–221. DOI: 10.1007/978-3-662-60651-3_8.

[32] N. Tax, X. Lu, N. Sidorova, D. Fahland, and W. M. P. van der Aalst. "The imprecisions of precision measures in process mining." In: *Information Processing Letters* 135 (2018), pp. 1–8. DOI: 10.1016/j.ipl.2018.01.013.

[33] S. K. vanden Broucke and J. De Weerdt. "Fodina: A robust and flexible heuristic process discovery technique." In: *Decision Support Systems* 100 (2017). Smart Business Process Management, pp. 109–118. DOI: https://doi.org/10.1016/j.dss.2017.04.005.

[34] X. Wan, V. Nguyen, H. Ha, B. Ru, C. Lu, and M. A. Osborne. "Think Global and Act Local: Bayesian Optimisation over High-Dimensional Categorical and Mixed Search Spaces." In: *Proceedings of the 38th International Conference on Machine Learning*. Vol. 139. Proceedings of Machine Learning Research. PMLR, July 2021, pp. 10663–10674. URL: https://proceedings.mlr.press/v139/wan21b.html.

[35] J. M. E. M. van der Werf, B. F. van Dongen, C. A. J. Hurkens, and A. Serebrenik. "Process Discovery Using Integer Linear Programming." In: *Applications and Theory of Petri Nets*. Petri Nets 2008. Vol. 5062. LNCS. Berlin, Heidelberg: Springer, 2008, pp. 368–387. DOI: 10.1007/978-3-540-68746-7_24.

[36] J. M. E. M. van der Werf, B. F. van Dongen, C. A. J. Hurkens, and A. Serebrenik. "Process Discovery using Integer Linear Programming." In: *Fundamenta Informaticae* 94.3-4 (2009), pp. 387–412. DOI: 10.3233/FI-2009-136.