# Model-Independent Error Bound Estimation for Conformance Checking Approximation

Mohammadreza Fani Sani[1,2], Martin Kabierski[3], Sebastiaan J. van Zelst[4,1], and Wil M.P. van der Aalst[1,4]

[1]Process and Data Science Chair, RWTH Aachen University, Aachen, Germany
[2]Microsoft Development Center Copenhagen, Copenhagen, Denmark
[3]Department of Computer Science, Humboldt-Universität zu Berlin, Berlin, Germany
[4]Fraunhofer FIT, Birlinghoven Castle, Sankt Augustin, Germany
{fanisani,s.j.v.zelst,wvdaalst}@pads.rwth-aachen.de,
martin.kabierski@hu-berlin.de

**Abstract** Conformance checking techniques quantify correspondence between a process's execution and a reference process model using event data. Alignments, used for conformance statistics, are computationally expensive for complex models and large datasets. Recent studies show accurate approximations can be achieved by selecting subsets of model behavior. This paper presents a novel approach deriving error bounds for conformance checking approximation based on arbitrary activity sequences. The proposed approach allows for the selection of relevant subsets for improved accuracy. Experimental evaluations validate its effectiveness, demonstrating enhanced accuracy compared to traditional alignment methods.

**Keywords:** Process mining · Conformance checking approximation · Alignments · Edit distance · Instance selection · Sampling

## 1 Introduction

Conformance checking, a sub-field of process mining, assesses the alignment between a process model and recorded event data [1]. Alignments, an established class of conformance checking artifacts [2], quantify deviations between recorded process execution and the intended behavior modeled by the process model.

Information systems generate vast amounts of event data that require efficient analysis. This *big event data*, combined with complex process models, leads to long computation times for alignments, limiting their practical application. However, in many cases, obtaining an approximate value is sufficient for meaningful conclusions instead of exact alignment values. For instance, genetic process discovery [3], evaluating generations of *candidate process models* based on an event log requires impractical exact alignment results. Yet, determining if a newly generated process model improves alignment results is sufficient. Therefore, fast alignment approximation techniques with guaranteed error are valuable.

Various approaches for alignment approximation have been proposed recently [4–8]. In our previous work [4], we utilize subsets of the process model's behavior for alignment approximation [4]. Initially, we construct alignments for a subset of the process behavior and estimate the alignment cost for the remaining

traces based on these alignments and edit distances, providing bounds for the approximated costs [4]. The quality of these subset-based approximations depends on the selected subset of model behavior [4]. Thus, quantifying the quality of an approximation based on a chosen subset aids in identifying suitable approximation subsets [4]. This paper therefore supports these approaches by introducing a novel approach for quantifying the quality of alignment approximations, which improves the error bounds introduced in our previous work [4].

Fig. 1 presents a schematic overview of the proposed approach. A process model $M$ models a process $P$ that generates an event log $L$. Existing approaches compute *exact* or *approximate conformance checking results* by considering the language of the model $\mathcal{L}(M)$ (possibly infinite), or, a relevant finite subset thereof. Our method computes error bounds for alignment approximation using a proxy-set $\Omega$. From $\Omega$, we derive the relevant subset of process model behavior $\mathcal{L}_F(M)$ and use it to approximate alignment costs of traces in $L$. We also provide bounds on the introduced approximation error.

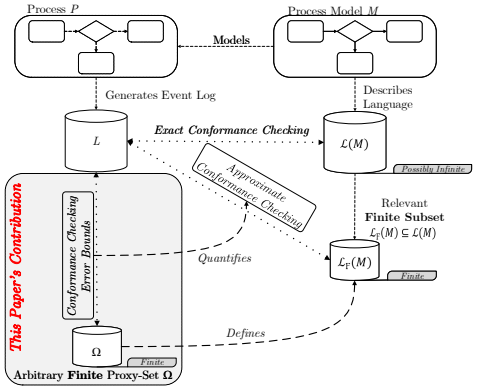



Figure 1: A process model $M$ represents a process $P$ generating an event log $L$. Existing approaches compute exact or approximate conformance checking results using the language of the model $\mathcal{L}(M)$. We propose quantifying error bounds for approximations by an arbitrary proxy-set $\Omega$.

We evaluate our new error bound estimation technique using real event logs. Our experiments confirm a correlation between the maximum error bounds calculated a-priori and the eventual approximation error. The accuracy improves by using more suitable subsets of process model behavior with lower error bounds. Additionally, the computation time for error bounds is negligible compared to exact alignments.

## 2 Related Work

Conformance checking techniques have been well-studied in the literature. In [1], different methods for conformance checking and its applications are covered. Alignments, introduced in [9] have rapidly developed into the standard conformance checking technique. In [10], decomposition technique is proposed for improving the performance of the alignment computation. In the context of stream-based process analytics, in [5] the authors propose to incrementally compute prefix-alignments.

Few papers consider the use of sampling in process mining. In [11], the authors recommend a trace-based statistical sampling method to decrease the required time for process discovery. Moreover, in [12], we analyzed random and

biased sampling methods with which we are able to adjust the size of the sampled data for process discovery.

Some research has focused on alignment approximation. General approximation schemes for alignments, i.e., the computation of near-optimal alignments, have been proposed in [13]. [6] proposes to incrementally sample the event log and check conformance on the sampled data. The approach incrementally increases the sample size until the approximated conformance value converges. The authors of [14] propose a conformance approximation method, that applies relaxation labeling methods on a partial order representation of a process model generated in a pre-processing step to produce alignments that are close to an optimal alignment. Furthermore, subset selection of model behaviors using instance selection [4] and simulation [15] have been proposed. The tool that supports these ideas is presented in [8]. In this context, in [7], the authors show, that a trie encoding of these selected subsets yields further runtime improvements.

## 3 Preliminaries

This section introduces conformance checking terminology and notation.

We let $\mathcal{B}(X)$ denote the set of all possible multisets over $X$. Given $b{\in}\mathcal{B}(X)$, $\overline{b}{=}\{x|b(x){>}0\}$. $X^*$ denotes the set of all sequences over $X$. Let $X'{\subseteq}X$ and let $\sigma{\in}X^*$, $\sigma_{\downarrow_{X'}}$ returns the projected sequence of $\sigma$ on set $X'$, e.g., $\langle a,b,c,b,d\rangle_{\downarrow_{\{b,d\}}}{=}\langle b,b,d\rangle$. Let $X_1, X_2, \ldots, X_n$ be $n$ arbitrary sets and let $X_1{\times}X_2{\cdots}{\times}X_n$ denote the corresponding Cartesian product. Let $\sigma{\in}(X_1{\times}X_2{\cdots}X_n)^*$ be a sequence of tuples, $\pi_i(\sigma)$ returns the sequence of elements in $\sigma$ at position $1{\leq}i{\leq}n$, e.g., $\pi_i(\langle(x_1^1,x_2^1,\ldots,x_n^1),(x_1^2,x_2^2,\ldots,x_n^2),\ldots,(x_1^{|\sigma|},x_2^{|\sigma|},\ldots,x_n^{|\sigma|})\rangle{=}\langle x_i^1,x_i^2,\ldots x_i^{|\sigma|}\rangle$.

Given $\sigma,\sigma'{\in}X^*$, $\delta(\sigma,\sigma'){\in}\mathbb{N}_{\geq 0}$ represents the *edit distance* (only using *insertions* and *deletions*) between $\sigma$ and $\sigma'$, i.e., the minimum number of edits required to transform $\sigma$ into $\sigma'$, e.g., $\delta(\langle w,x,y\rangle,\langle x,y,z\rangle){=}2$ (delete $w$ and add $z$). Note that $\delta(\sigma,\sigma'){=}\delta(\sigma',\sigma)$ ($\delta$ is symmetrical) and $\delta(\sigma,\sigma''){\leq}\delta(\sigma,\sigma'){+}\delta(\sigma',\sigma'')$ (triangle inequality applies to $\delta$). Given a sequence $\sigma{\in}X^*$ and a set of sequences $S{\subseteq}X^*$, we define $\Delta(\sigma,S){=}\min_{\sigma'\in S}\delta(\sigma,\sigma')$.

Event logs are collections of events that represent the execution of multiple process instances. They serve as the foundation for process mining algorithms. These events capture the timing of activities, denoted by their starting and finishing times, for each instance of the process identified by *Case*. In certain cases, such as alignment computation, only the control-flow information, which refers to the sequences of executed activities within a process instance, is necessary. Thus, we utilize the aforementioned mathematical model of an event log.

**Definition 1 (Event Log).** *Let $\Sigma$ denote the* universe of activities. *A* trace $\sigma$ *is a sequence of activities ($\sigma{\in}\Sigma^*$). An* event log $L{\in}\mathcal{B}(\Sigma^*)$ *is a bag of traces.*

Process models are used to describe the behavior of a process. They can take the form of simple conceptual drawings or more complex mathematical concepts such as Petri nets and BPMN diagrams. An example of a process model is shown

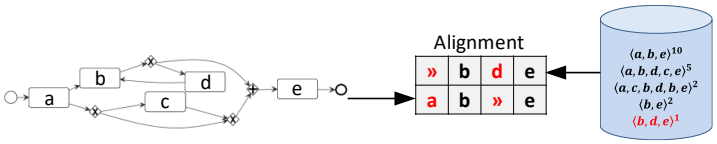



Figure 2: A process model $M_1$ and an event log $L_1$. The optimal alignment of the last trace of $L_1$ and $M_1$ is shown in the middle of the figure.

in Fig. 2 that uses BPMN notation. In this paper, we do not assume a specific modeling notation, but rather that process models describe activity sequences.

**Definition 2 (Process Model).** *Let $\Sigma$ denote the universe of activities. A process model $M$ describes the intended behavior of a process. We refer to the behavior described by model $M$ as its* language $\emptyset \subset \mathcal{L}(M) \subseteq \Sigma^*$*, i.e., a collection of activity sequences.*

For $M_1$ in Fig. 2, we have $\mathcal{L}(M_1)=\{\langle a,b,e\rangle, \langle a,b,c,e\rangle, \langle a,c,b,e\rangle, \langle a,b,d,b,e\rangle, \ldots\}$. Due to the existence of loops, the language of a process model may be infinite.

To quantify whether an event log conforms to a process model, we use alignments. An alignment between a trace and a model describes which events in the trace can be "aligned with activities described by the process model". Furthermore, alignments indicate whether an event cannot be explained by the model or whether an activity as described by the model was not observed. In Fig. 2, an alignment of trace $\langle b,d,e\rangle$, and the given process model is provided. Observe that the trace does not contain activity $a$, which should always be present according to the model. In the alignment, this is visualized by the first column $\frac{\gg}{a}$. Similarly, after the observed $d$-activity, no second $b$-activity was observed. As such, in this alignment, the occurrence of $d$ is rendered obsolete, i.e., visualized as $\frac{d}{\gg}$. We formally define alignments as follows.

**Definition 3 (Alignment).** *Let $\Sigma$ denote the universe of activities, let $M$ be a process model with corresponding language $\emptyset \subset \mathcal{L}(M) \subseteq \Sigma^*$ and let $\sigma \in \Sigma^*$ be a trace. An alignment $\gamma$ of $\sigma$ and $M$, is a sequence, characterized as $\gamma \in ((\Sigma \cup \{\gg\}) \times (\Sigma \cup \{\gg\}))^*$, s.t., $\pi_1(\gamma)_{\downarrow_\Sigma} = \sigma$ and $\pi_2(\gamma)_{\downarrow_\Sigma} \in \mathcal{L}(M)$. The set of all possible alignments of trace $\sigma$ and model language $\mathcal{L}(M)$ is denoted as $\Gamma(\sigma, \mathcal{L}(M))$.*

Let $c\colon (\Sigma \cup \{\gg\}) \times (\Sigma \cup \{\gg\}) \to \mathbb{R}$ be an arbitrary cost function, assigning costs to the different type of alignments moves, then, given $\sigma \in \Sigma^*$, $M \subseteq \Sigma^*$ and $\gamma \in \Gamma(\sigma, \mathcal{L}(M))$, we let $\kappa_c(\gamma) = \sum_{1 \leq i \leq |\gamma|} c(\gamma(i))$ denote the cost of alignment $\gamma$. We let $\Gamma_c^\star(\sigma, \mathcal{L}(M)) = \underset{\gamma \in \Gamma(\sigma, \mathcal{L}(M))}{\arg\min} \kappa_c(\gamma)$ be the set of *optimal/minimal alignments*, i.e. the set of alignments, whose corresponding cost under the given cost function is minimal, and $z_c(\sigma, \mathcal{L}(M)) = \min_{\gamma \in \Gamma(\sigma, \mathcal{L}(M))} \kappa_c(\gamma)$ be the optimal alignment cost for trace $\sigma$ and model $M$ (hence: $\forall \gamma \in \Gamma_c^\star(\sigma, \mathcal{L}(M))\,(\kappa_c(\gamma) = z_c(\sigma, \mathcal{L}(M)))$). In the context of this paper, given $\gamma \in \Gamma_c^\star(\sigma, \mathcal{L}(M))$, we write $\varphi(\gamma) = \pi_2(\gamma)_{\downarrow_\Sigma}$ to refer to *the "model behavior" corresponding to* $\sigma$, i.e., the projection of $\sigma$ onto any of the closest possible execution sequence in $M$.

In the remainder, we assume that $c$ represents the *standard cost function*, i.e., $\forall a \in \Sigma$, $c(a, \gg) = c(\gg, a) = 1$, $c(a,a) = 0$, and $c(a,a') = \infty$ if $a \neq a'$, and we omit it as a subscript.

## 4 Estimating Alignment Error Bounds

In this section, we derive error bounds for proxy sets $\Omega$ (Section 4.1). The *edit distance* between sequences provides upper and lower bounds for trace and model alignment costs. We also approximate optimal proxy sets that minimize cumulative approximation error (Section 4.2). Finally, we discuss enhancements to the bounds (Section 4.3).

### 4.1 Computing the Maximal Alignment Approximation Error

Here, we show that for given traces $\sigma, \sigma' \in \Sigma^*$ and a model $M$, the edit distance $\Delta(\sigma, \sigma')$ gives a range for the actual optimal alignment value $z(\sigma, \mathcal{L}(M))$. First, we show that under the standard cost function, we can use the edit distance for computing the cost of the optimal alignment between two arbitrary sequences.

**Lemma 1 (Edit Distance Quantifies Optimal Alignment Costs).** *Let $\Sigma$ denote the universe of activities, let $\sigma \in \Sigma^*$ be a trace, let $M$ be a process model and let $\gamma \in \Gamma^\star(\sigma, \mathcal{L}(M))$ be an optimal alignment of $\sigma$ and $M$. Using the standard cost function, $\kappa(\gamma) = \delta(\sigma, \varphi(\gamma))$*

*Proof.* *$\gamma$ only contains $(a, a)$, $(a, \gg)$, and $(\gg, a)$ elements. Let $R$ be the set of $(a, \gg)$ elements and $I$ be the set of $(\gg, a)$ elements. Converting $\sigma$ into $\varphi(\gamma)$ is done by* removing *activities in $\sigma$ and inserting activities represented by $R$ and $I$, respectively. Thus, $\kappa(\gamma) = R + I$. Similarly, $\delta(\sigma, \varphi(\gamma))$ indicates the minimum number of insertions/removals to transform $\sigma$ into $\varphi(\gamma)$. If $\kappa(\gamma) < \delta(\sigma, \varphi(\gamma))$, then $\delta(\sigma, \varphi(\gamma))$ does not represent the minimal number of edits. Likewise, if $\kappa(\gamma) > \delta(\sigma, \varphi(\gamma))$, then $\gamma$ is not optimal.*

**Corollary 1 ($\Delta(\sigma, \mathcal{L}(M))$ equals $z(\sigma, \mathcal{L}(M))$).** *Let $\Sigma$ denote the universe of activities, let $\sigma \in \Sigma^*$ be a trace, let $M$ be a process model with corresponding language $\emptyset \subset \mathcal{L}(M) \subseteq \Sigma^*$. Using the standard cost function, $z(\sigma, \mathcal{L}(M)) = \Delta(\sigma, \mathcal{L}(M))$.*

*Proof.* *Let $\gamma \in \Gamma^\star(\sigma, \mathcal{L}(M))$, then, $z(\sigma, \mathcal{L}(M)) = \kappa(\gamma) = \delta(\sigma, \varphi(\gamma)) = \Delta(\sigma, \mathcal{L}(M))$.*

Again, assume the two traces to align to be $\sigma = \langle b, d, e \rangle$ and $\varphi(\sigma) = \langle a, b, e \rangle$ from Fig. 2. It is easy to see, that the edit distance of the two traces is 2 (insertion of $a$ and deletion of $b$ in $\sigma$), which is equivalent to the alignment cost.

Now, we show that, given an arbitrary sequence with known alignment cost, we can derive bounds for the possible alignment cost of another activity sequence. This allows the approximation of said cost without relying on the construction of alignments.

**Theorem 1 (Edit Distance Provides Approximation Bounds).** *Let $\sigma, \sigma' \in \Sigma^*$ be two traces and let $M$ be a process model with corresponding language $\emptyset \subset \mathcal{L}(M) \subseteq \Sigma^*$. The optimal alignment value $z(\sigma, \mathcal{L}(M))$, is within $\delta(\sigma, \sigma')$ of $z(\sigma', \mathcal{L}(M))$, i.e., $z(\sigma', \mathcal{L}(M)) - \delta(\sigma, \sigma') \leq z(\sigma, \mathcal{L}(M)) \leq z(\sigma', \mathcal{L}(M)) + \delta(\sigma, \sigma')$.*

*Proof.* *Let $\gamma \in \Gamma^\star(\sigma, \mathcal{L}(M))$ and let $\gamma' \in \Gamma^\star(\sigma', \mathcal{L}(M))$. Triangle inequality of edit distance yields $\delta(\sigma, \varphi(\gamma')) \leq \delta(\sigma, \sigma') + \delta(\sigma', \varphi(\gamma'))$, which we can rewrite ( Lemma 1) to $\delta(\sigma, \varphi(\gamma')) \leq \delta(\sigma, \sigma') + z(\sigma', \mathcal{L}(M))$. Since $z(\sigma, \mathcal{L}(M)) \leq \delta(\sigma, \varphi(\gamma'))$, we have: $z(\sigma, \mathcal{L}(M)) \leq z(\sigma', \mathcal{L}(M)) + \delta(\sigma, \sigma')$.*

*Similarly,* $\delta(\sigma', \varphi(\gamma)) \leq \delta(\sigma, \sigma') + \delta(\sigma, \varphi(\gamma))$. *We deduce* $\delta(\sigma', \varphi(\gamma)) \leq \delta(\sigma, \sigma') + z(\sigma, \mathcal{L}(M))$. *As* $z(\sigma', \mathcal{L}(M)) \leq \delta(\sigma', \varphi(\gamma))$, *we deduce* $z(\sigma', \mathcal{L}(M)) - \delta(\sigma, \sigma') \leq z(\sigma, \mathcal{L}(M))$. *Hence, we obtain* $z(\sigma', \mathcal{L}(M)) - \delta(\sigma, \sigma') \leq z(\sigma, \mathcal{L}(M)) \leq z(\sigma', \mathcal{L}(M)) + \delta(\sigma, \sigma')$.

In Fig. 2, $z(\langle a, c, c, b, d, e\rangle, \mathcal{L}(M_1)) = 2$ and $\delta(\langle a, c, c, b, d, e\rangle, \langle a, c, b, d, e\rangle) = 1$. We deduce $1 \leq z(\langle a, c, b, d, e\rangle, \mathcal{L}(M_1)) \leq 3$. If $z(\langle a, c, c, b, d, e\rangle, \mathcal{L}(M_1))$ is unknown, $\delta(\langle a, c, c, b, d, e\rangle, \langle a, c, b, d, e\rangle) = 1$ implies that using it for approximating $z(\langle a, c, b, d, e\rangle, \mathcal{L}(M_1))$ *yields a maximal absolute approximation error of* 1.

### 4.2 Generating Proxy-Sets

Theorem 1 implies that, given a process model $M$ and traces $\sigma, \sigma' \in \Sigma^*$, when using $z(\sigma', \mathcal{L}(M))$ for approximating $z(\sigma, \mathcal{L}(M))$, we obtain an approximation error $\epsilon \leq \delta(\sigma, \sigma')$, i.e. the maximum approximation error is $\delta(\sigma, \sigma')$. Interestingly, the error bounds on $\epsilon$ is determined independently of the model. Furthermore, $\sigma'$ is allowed to be an arbitrary sequence, i.e., it is perfectly fine if $\sigma' \notin \mathcal{L}(M)$, and, given some $L \in \mathcal{B}(\Sigma^*)$ s.t. $\sigma \in \overline{L}$, $\sigma' \notin \overline{L}$. Hence, given an arbitrary set of sequences $\Omega \subseteq \Sigma^*$, $\arg\min_{\sigma' \in \Omega} \delta(\sigma, \sigma')$ represents the members of $\Omega$ that minimize the expected maximum error when using $z(\sigma', \mathcal{L}(M))$ for approximating (i.e., for $\sigma' \in \arg\min_{\sigma' \in \Omega} \delta(\sigma, \sigma')$).

For an event log $L \in \mathcal{B}(\Sigma^*)$ and proxy-set $\Omega \subseteq \Sigma^*$, $\forall \sigma \in \overline{L} \left( \min_{\sigma' \in \Omega} \delta(\sigma, \sigma') = 0 \right) \Leftrightarrow \Omega \supseteq \overline{L}$, i.e., if every member of the log has an edit distance of 0 w.r.t. the proxy-set, then every member of the event log is a member of the proxy-set. Clearly, in such a case, using proxy-set $\Omega$ yields optimal alignments, yet, at the same (or even worse) time and memory complexity as computing conventional optimal alignments.

In the remainder, given an event log $L \in \mathcal{B}(\Sigma^*)$ and proxy-set $\Omega \subseteq \Sigma^*$, let $\epsilon_\Omega(L) = \sum_{\sigma \in \overline{L}} L(\sigma) \cdot \min_{\sigma' \in \Omega} \delta(\sigma, \sigma')$ be the accumulative approximation error of $L$ using $\Omega$. Given two proxy-sets $\Omega, \Omega' \subseteq \Sigma^*$, $\Omega$ *dominates* $\Omega'$ for event log $L$ if and only if $\epsilon_\Omega(L) \leq \epsilon_{\Omega'}(L)$ and $|\Omega| < |\Omega|'$ and we refer to $\Omega'$ as a *redundant* proxy-set. A proxy-set $\Omega$ is *k-optimal* for event log $L$ if and only if $\forall \Omega' \in \Sigma^* \, (|\Omega'| = k \implies \epsilon_\Omega(L) \leq \epsilon_{\Omega'}(L))$. A *k-optimal* proxy-set $\Omega$ is *k-primal* if $|\Omega| = k$. For example, $\Omega = \overline{L}$ is $|\overline{L}|$-primal, 1-optimal, 2-optimal, ..., $|\overline{L}|$-optimal. Furthermore, it is easy to see that any ($k$-primal) proxy-set $\Omega$ with $|\Omega| > L$ is dominated by $L$ and hence redundant. More interestingly, primal proxy-sets that are smaller than the event log are never redundant.

**Theorem 2 (Primal Proxy-Sets are Non-Redundant).** *Let* $L \in \mathcal{B}(\Sigma^*)$ *be an event log,* $\Omega \subseteq \Sigma^*$ *be a proxy-set such that* $|\Omega| < |\overline{L}|$, *and* $\Omega$ *is* k-primal. $\Omega$ *is* non-redundant.

*Proof. Assume that* $\Omega$ *is redundant. Hence,* $\exists \Omega' \subseteq \Sigma^* \, (|\Omega'| < |\Omega| \wedge \epsilon_{\Omega'}(L) \leq \epsilon_\Omega(L))$. *However, observe that, we are able to create* $\Omega'' = \Omega' \cup L''$ *with* $|L''| = |\Omega| - |\Omega'|$ *and* $\sigma \in L'' \implies \sigma \in \overline{L} \wedge \sigma \notin \Omega'$ *(note that* $|\Omega| = |\Omega''|$*). Observe that* $\epsilon_{\Omega''}(L) < \epsilon_{\Omega'}(L)$ *and as a consequence* $\epsilon_{\Omega''}(L) < \epsilon_\Omega(L)$, *contradicting the fact that* $\Omega$ *is k-primal.*

Note, Theorem 2 implies the existence of a $k$-primal proxy-set $\Omega$ for any event log $L \in \mathcal{B}(\Sigma^*)$ and $k \in 1, 2, \ldots, |L|$. This $k$-primal proxy-set minimizes the accumulative approximation error $\epsilon_\Omega(L)$ for size $k$ and can be considered the optimal proxy-set for that size. However, finding such proxy-sets is an NP-Hard problem and goes beyond the scope of this paper. Instead, here we focus on proxy-sets where $\Omega \subseteq L$. In the following paragraphs, we introduce different methods for generating proxy-sets and their relation to the optimal primal proxy-sets.

**Sampling** Proxy-sets can be generated using sampling methods, either directly from the event log, the given process model, or a mixture thereof. In previous work, we investigated the sampling of model behavior using uniform distributions [4] and event-log-guided process model simulation [15].

Strictly sampling the behavior from the process model, i.e., $\Omega \subseteq \mathcal{L}(M)$, particularly when using event log-guided simulation yields (under standard cost function) $z(\sigma', \mathcal{L}(M)) = 0,\ \forall \sigma' \in \Omega$, and thus $0 \leq z(\sigma, \mathcal{L}(M)) \leq \Delta(\sigma, \sigma'), \forall \sigma \in L$. While it is very unlikely that such a proxy-set is $k$-primal due to it being closer to the log behaviour, $z(\sigma', \mathcal{L}(M)) = 0,\ \forall \sigma' \in \Omega$, can be exploited.

Sampling $\Omega$ from the event log is likely to result in a proxy-set that is closer to a $k$-primal solution, especially when prioritizing $\sigma \in \overline{L}$, for which $L(\sigma)$ is high. Hence, using log-based sampling is more likely to minimize $\epsilon_\Omega$. However, since the actual $z(\sigma', \mathcal{L}(M))$ for $\sigma' \in \Omega$ is unknown, we cannot tighten the estimator.

**Centroid-Based Clustering** For a given target size $k$, the optimal proxy-set is *k-primal*. As an alternative to sampling, *clustering algorithms* are suitable for proxy-set selection. These algorithms group objects into clusters based on their similarity or distance, often using the edit distance as a metric. *Centroid-based clustering algorithms*, such as *K-Medoids* [16], are particularly relevant as they assign objects to the centroid with the minimal distance. While clustering algorithms can be applied to any set of activity sequences, applying them to the input event log produces proxy-sets close to the *k-primal* solution. Due to the time-consuming nature of providing optimal clustering solutions, several faster approximation techniques have been proposed.

### 4.3 Improving the Alignment Approximation Bounds

We showed that $\Omega$ and proxy-sequence $\sigma' \in \Omega$ can quantify the approximation error $\epsilon$ as $\epsilon \leq \delta(\sigma, \sigma')$ when approximating $z(\sigma, \mathcal{L}(M))$ with $z(\sigma', \mathcal{L}(M))$. Now, we show how using proxy-sets can improve alignment approximation bounds.

When approximating alignments using $\Omega$, we first compute the alignments of $\Omega$ traces. We derive the bounds of the alignment cost of $z(\sigma, \mathcal{L}(M))$ by simply adding/subtracting $\delta(\sigma, \sigma')$ to $z(\sigma', \mathcal{L}(M))$. Note, when using the standard cost function, the lower bound of any alignment cannot be lower than 0. In certain cases, we can derive a tighter lower bound. Let $\Sigma_M = \{a \in \Sigma | \exists \sigma \in \mathcal{L}(M)(a \in \sigma)\}$, then, for any $\sigma \in \Sigma^*$, $z(\sigma, \mathcal{L}(M)) \geq |\sigma_{\downarrow_{\Sigma \setminus \Sigma_M}}|$, i.e., the elements of $\sigma_{\downarrow_{\Sigma \setminus \Sigma_M}}$ are always moves of the form $\frac{a}{\gg}$. Furthermore, in case $|\sigma| < \min_{\sigma' \in \mathcal{L}(M)} |\sigma'|$, we need at

Table 1: Statistics regarding the real event logs that are used in the experiment.

| Event Log | Activities | Traces | Variants |
|---|---|---|---|
| *BPIC-2012* | 23 | 13087 | 4336 |
| *BPIC-2018-Inspection* | 15 | 5485 | 3190 |
| *BPIC-2019* | 42 | 251734 | 11973 |
| *Hospital-Billing* | 18 | 100000 | 1020 |
| *Road* | 11 | 150370 | 231 |
| *Sepsis* | 16 | 1050 | 846 |

least $|\sigma'|-|\sigma|$ (where $\sigma' \in \underset{\sigma' \in \mathcal{L}(M)}{\arg\min} |\sigma'|$) moves of the form $\frac{\gg}{a}$. Hence, the theoretical lower-bound of any $\sigma \in \Sigma^*$ is equal to $\max(0, \underset{\sigma' \in \mathcal{L}(M)}{\min}(|\sigma'|)-|\sigma|)+|\sigma_{\downarrow_{\Sigma \setminus \Sigma_M}}|$. We correspondingly define the $\Omega$-driven lower and upper bound as follows.

**Definition 4 ($\Omega$-Driven Alignment Bounds).** *Let $\Sigma$ denote the universe of activities, let $M$ be a process model with corresponding language $\emptyset \subset \mathcal{L}(M) \subseteq \Sigma^*$ and let $\Omega \subseteq \Sigma^*$ be a proxy-set. Let $\top_{\Omega,M} : \Sigma^* \to \mathbb{N}$ denote the $\Omega$-driven upper bound and $\bot_{\Omega,M} : \Sigma^* \to \mathbb{N}$ the $\Omega$-driven lower bound, s.t.:*

$$\top_{\Omega,M}(\sigma) = \min_{\sigma' \in \Omega}(z(\sigma', \mathcal{L}(M)) + \delta(\sigma, \sigma')) \tag{1}$$

$$\bot_{\Omega,M}(\sigma) = \max(\max(0, \min_{\sigma' \in \mathcal{L}(M)}(|\sigma'|)-|\sigma|)+|\sigma_{\downarrow_{\Sigma \setminus \Sigma_M}}|, \max_{\sigma' \in \Omega}(z(\sigma', \mathcal{L}(M)) - \delta(\sigma, \sigma'))) \tag{2}$$

Finally, given $\top_{\Omega,M}$ and $\bot_{\Omega,M}$, we quantify the approximated alignment cost of $\sigma \in \Sigma^*$ as the average of bounds to minimize the possible approximation error, i.e., $\hat{z}_\Omega(\sigma, \mathcal{L}(M))$, as $\hat{z}_\Omega(\sigma, \mathcal{L}(M)) = \frac{\top_{\Omega,M}(\sigma) - \bot_{\Omega,M}(\sigma)}{2}$. In theory, it is possible to assign different weights to bounds based on additional knowledge or bias.

## 5 Evaluation

To assess the efficacy of the proposed error bounds, we conducted an extensive evaluation using multiple publicly available event logs. In particular, we explored the accuracy and the runtime performance of the proposed bounds. First, we briefly describe the implementation and evaluation setup(Section 5.1), followed by a discussion of the evaluation results(Section 5.2)

### 5.1 Experimental Setup

To evaluate the proposed error bounds, we implemented the *Conformance Approximation* plug-in in the `ProM` [17] framework[1], including various proxy-set generation methods (cf. Section 4.2).

The proposed methods were applied to six real event logs, and basic information about these logs, such as the number of distinct activities, traces, and variants, is provided in Table 1. For each event log, we apply conformance checking using process models obtained via the Inductive Miner algorithm [18]

[1] `svn.win.tue.nl/repos/prom/Packages/LogFiltering`

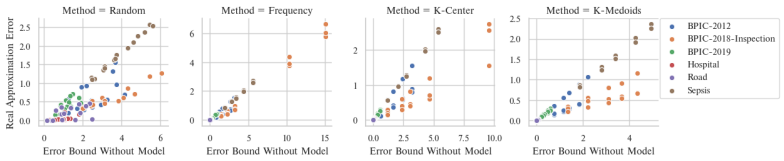



Figure 3: Scatter plots of the maximum approximation error and the real approximation error using different proxy-set generation methods.

with infrequent thresholds of 0.2, 0.4, and 0.6. Four proxy-set generation methods are used:*random sampling*, *frequency-based sampling*, *K-Medoids clustering* and *K-Center clustering*. In random sampling, variants are uniformly sampled (without replacement) from the event log. In frequency-based sampling, traces are selected based on their $L(\sigma)$-values in descending order. K-Medoids clustering determines centroids by minimizing pairwise dissimilarity between traces, while K-Center clustering minimizes the maximum distance between centroids and traces. Proxy-set sizes were varied using different percentages (5%, %10, 20%, 30%, 50%) of the number of variants in the event logs. Each experiment was repeated four times.

### 5.2 Results

First, we analyze the relationship between maximum and actual approximation error. Next, we examine the time performance of the estimation. Finally, we assess the effectiveness of the proposed lower bound.

**Maximum Approximation Error versus Approximation Error** Observe that minimizing the expected maximum error, e.g., by selecting a seemingly optimal proxy set, does not guarantee a minimal approximation error. For example, given some model $M$, $\sigma{\in}\Sigma^*$, $\Omega{=}\{\sigma_1,\sigma_2\}$ and $\Omega'{=}\{\sigma_1,\sigma_3\}$, assume that $\delta(\sigma,\sigma_1){=}2$, $\delta(\sigma,\sigma_2){=}3$ and $\delta(\sigma,\sigma_3){=}1$. Clearly, the maximal error based on $\Omega$ is 2, and, based on $\Omega'$, it is 1. As such, we intuitively favor $\Omega'$ over $\Omega$. However, if $z(\sigma_1,\mathcal{L}(M)){=}7$, $z(\sigma_2,\mathcal{L}(M)){=}2$ and $z(\sigma_1,\mathcal{L}(M)){=}6$, we obtain $\bot_{\Omega,M}(\sigma){=}\top_{\Omega,M}(\sigma){=}5$, whereas $\bot_{\Omega',M}(\sigma){=}5$ and $\top_{\Omega',M}(\sigma){=}7$. Hence, from $\Omega$, we derive that $z(\sigma,\mathcal{L}(M)){=}5$ (note $\hat{z}_{\Omega}(\sigma,\mathcal{L}(M)){=}5$), whereas from $\Omega'$, we derive $5{\leq}z(\sigma,\mathcal{L}(M)){\leq}7$ (with $\hat{z}_{\Omega'}(\sigma,\mathcal{L}(M)){=}6$). Thus, utilizing $\Omega$ gives the exact alignment value, whereas using $\Omega'$ yields an error of 1.

Given that there is no causal relation between the maximum approximation error and the actual error, we investigate, the strength of the correlation between the maximum approximation error and the effective approximation for each of the proposed proxy-set generation methods. The scatter plots in Figure 3 illustrate these values for each method, distinguishing event logs with different colors.

Table 2: Pearson correlation coefficients between the maximum approximation error and the real approximation errors for different methods.

| Log Name | Random | Frequency | K-Center | K-Medoids |
|---|---|---|---|---|
| BPIC-2012 | 0.577 | 0.701 | 0.820 | 0.806 |
| BPIC-2018-Inspection | 0.933 | 0.945 | 0.995 | 0.862 |
| BPIC-2019 | 0.682 | 0.842 | 0.998 | 0.843 |
| Hospital | 0.534 | 0.782 | 0.969 | 0.962 |
| Road | 0.468 | 0.631 | 0.997 | 0.998 |
| Sepsis | 0.997 | 0.990 | 0.998 | 0.994 |

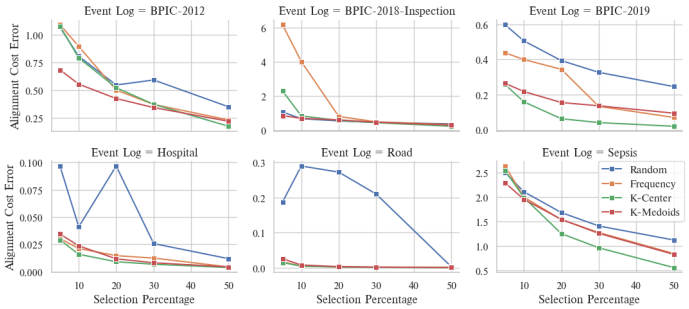


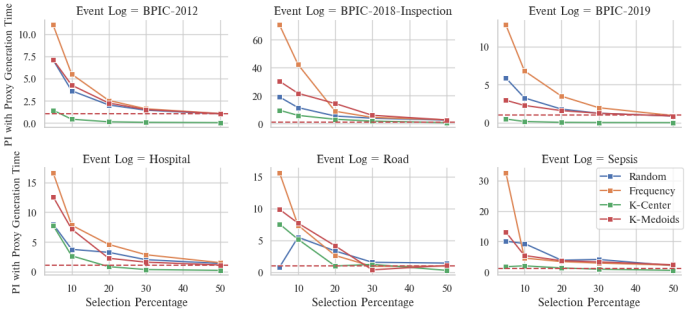

Figure 4: Effect of increasing the selected percentage of variants on approximated alignments' accuracy for different methods.

Moreover, we present the Pearson correlation coefficients in Table 2. The K-Center method demonstrates the highest correlation across all event logs. Notably, strong correlations between the maximum approximation error and the effective approximation error are observed for frequency-based sampling, K-Center, and K-Medoids. In contrast, random sampling exhibits a weaker correlation, especially for the Hospital-Billing and Road logs, where representative variants are limited, and random sampling fails to prioritize them.

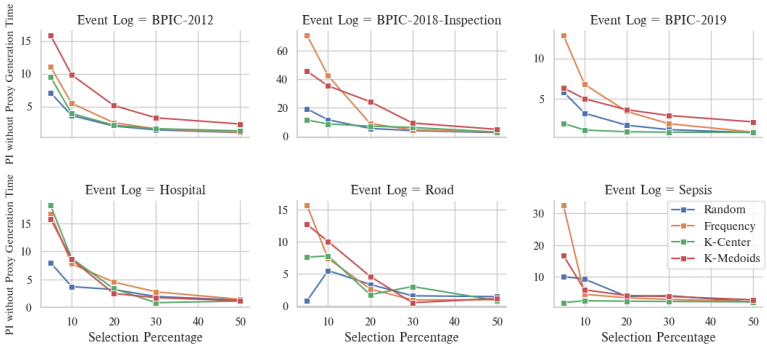
In Fig. 4, we demonstrate the impact of various proxy-set generation methods and trace variant percentages on approximated alignment cost accuracy. K-Center and K-Medoids show promising results, producing proxy-sets that improve accuracy. Additionally, larger proxy-set sizes reduce alignment cost errors, although the influence is constrained for similar event log variants.

**Conformance Checking Performance Improvement** We evaluated the time performance of the proxy-set generation methods and observed performance improvements in conformance checking (Fig. 5). To compute the *performance improvement* $PI$, we divide the conventional alignment computation time by the alignment approximation time, including and excluding proxy-set generation time. Higher PI values indicate greater performance improvement, while a PI value less than 1 indicates additional overhead. The frequency-based method shows the greatest improvement, as it quickly selects variants for proxy-set generation. The Random method has a lower PI value as it may select variants that require more time for alignment computation. Increasing the proxy-set size reduces performance gains. In some cases, the performance does not improve when considering proxy generation time. Thus, it is important to avoid selecting too many traces as a proxy. The proxy generation time for K-Center and K-Medoids methods is notably higher, especially for larger proxy-set sizes. However, if we separate the proxy generation time (as explained in Section 1), we can still improve the efficiency of the conformance checking procedure.

**Efficiency of the Proposed Lower Bound** Finally, in the last experiment, we compare the lower bound approximation without $M'$, i.e., $\max(0, \min_{\sigma' \in \mathcal{L}(M)} (|\sigma'|) - |\sigma|) + |\sigma_{\downarrow_{\Sigma \setminus \Sigma_M}}|$ and the lower bound that incorporates $M'$, i.,e., $\max_{\sigma' \in \Omega} (z(\sigma', \mathcal{L}(M)) - \delta(\sigma, \sigma'))$.




(a) Performance improvement with consideration of proxy-set generation time.




(b) Performance improvement without consideration of proxy selection time.

Figure 5: Impact of variant selection and proxy methods on performance improvement.

Table 3 presents the percentages of traces with higher values using various bounds. When both methods yield the highest value, we acknowledge both. The findings suggest that, in the majority of situations, employing the proposed lower bound derived from the proxy-set and its alignments is satisfactory. This approach yields more precise approximations of error bounds, leading to more informative evaluations of alignment cost approximations with a given proxy-set.

Table 3: Average of times that lower bounds have the highest value.

| Log Name | Without Ω | With Ω |
|---|---|---|
| BPIC-2012 | 80% | 100% |
| BPIC-2018- | 66% | 62% |
| BPIC-2019 | 95% | 99% |
| Hospital | 50% | 100% |
| Road | 87% | 100% |
| Sepsis | 100% | 100% |

## 6 Conclusion

In this paper, we proposed a method to obtain bounds on the approximation error when alignment costs are approximated using a subset of traces. Evaluations on real event logs validate the accuracy of different non-optimal instance selection methods and the proposed error estimation technique, and show a reduction in error for the approximated alignment costs while reducing the runtime. We aim to enhance the derivation of k-primal proxy sets to minimize approximation error and improve selection strategies. This will lead to more accurate alignment cost approximations, making our methodology valuable for process mining and alignment analysis.

## References


1. Carmona, J., van Dongen, B., Solti, A., Weidlich, M.: Conformance Checking. Springer (2018)
2. Adriansyah, A., Munoz-Gama, J., Carmona, J., van Dongen, B., van der Aalst, W.M.P.: Alignment based Precision Checking. In: International Conference on Business Process Management, Springer (2012) 137–149
3. Buijs, J.C., van Dongen, B., van der Aalst, W.M.P.: On the Role of Fitness, Precision, Generalization and Simplicity in Process Discovery. In: OTM, " On the Move to Meaningful Internet Systems", Springer (2012) 305–322
4. Fani Sani, M., van Zelst, S.J., van der Aalst, W.M.P.: Conformance checking approximation using subset selection and edit distance. In: 32nd International Conference, CAiSE, France, 2020, Proceedings, Springer 234–251
5. van Zelst, S.J., Bolt, A., Hassani, M., van Dongen, B.F., van der Aalst, W.M.: Online conformance checking: relating event streams to process models using prefix-alignments. International Journal of Data Science and Analytics (2017) 1–16
6. Bauer, M., van der Aa, H., Weidlich, M.: Sampling and approximation techniques for efficient process conformance checking. Information Systems (2020) 101666
7. Awad, A., Raun, K., Weidlich, M.: Efficient approximate conformance checking using trie data structures. In: 3rd International Conference on Process Mining (ICPM). (2021) 1–8
8. Fani Sani, M., Gonzalez, J.J.G., van Zelst, S.J., van der Aalst, W.M.P.: Alignment approximator: A prom plug-in to approximate conformance statistics. In: Proceedings of the Best Dissertation Award, Doctoral Consortium, and Demonstration & Resources Forum at BPM 2023, Utrecht, The Netherlands, 2023. (2023) 102–106
9. van der Aalst, W.M.P., Adriansyah, A., van Dongen, B.F.: Replaying history on process models for conformance checking and performance analysis. Wiley Interdiscip. Rev. Data Min. Knowl. Discov. **2**(2) (2012) 182–192
10. van der Aalst, W.M.P.: Decomposing petri nets for process mining: A generic approach. Distributed and Parallel Databases **31**(4) (2013) 471–507
11. Bauer, M., Senderovich, A., Gal, A., Grunske, L., Weidlich, M.: How much event data is enough? A statistical framework for process discovery. In: International Conference on Advanced Information Systems Engineering. (2018) 239–256
12. Fani Sani, M., van Zelst, S.J., van der Aalst, W.M.P.: Improving the performance of process discovery algorithms by instance selection. Comput. Sci. Inf. Syst. **17**(3) (2020) 927–958
13. Taymouri, F., Carmona, J.: A recursive paradigm for aligning observed behavior of large structured process models. In: International Conference on Business Process Management, Springer (2016) 197–214
14. Padró, L., Carmona, J.: Computation of alignments of business processes through relaxation labeling and local optimal search. Information Systems **104** (2022)
15. Fani Sani, M., Garza Gonzalez, J.J., van Zelst, S.J., van der Aalst, W.M.P.: Conformance checking approximation using simulation. In: 2nd International Conference on Process Mining, ICPM 2020, Padua, Italy, IEEE (2020) 105–112
16. Park, H., Jun, C.: A simple and fast algorithm for k-medoids clustering. Expert Syst. Appl. **36**(2) (2009) 3336–3341
17. van der Aalst, W.M.P., van Dongen, B., Günther, C.W., Rozinat, A., Verbeek, E., Weijters, T.: Prom: The process mining toolkit. BPM (Demos) **489**(31) (2009)
18. Leemans, S.J., Fahland, D., van der Aalst, W.M.P.: Discovering Block-Structured Process Models from Event Logs Containing Infrequent Behaviour. In: BPI. (2014) 66–78