Automatic Root Cause Identification Using Most Probable Alignments

Marie Koorneef[1], Andreas Solti[2], Henrik Leopold[1], Hajo A. Reijers[1,3]

[1] Department of Computer Sciences, Vrije Universiteit Amsterdam, The Netherlands
[2] Institute for Information Business, Vienna University of Economics and Business, Austria
[3] Department of Mathematics and Computer Science, Eindhoven University of Technology, The Netherlands

Abstract. In many organizational contexts, it is important that behavior conforms to the intended behavior as specified by process models. Non-conforming behavior can be detected by aligning process actions in the event log to the process model. A probable alignment indicates the most likely root cause for non-conforming behavior. Unfortunately, available techniques do not always return the most probable alignment and, therefore, also not the most probable root cause. Recognizing this limitation, this paper introduces a method for computing the *most probable alignment*. The core idea of our approach is to use the history of an event log to assign probabilities to the occurrences of activities and the transitions between them. A theoretical evaluation demonstrates that our approach improves upon existing work.

Keywords: conformance checking, root cause analysis, most probable alignments

1 Introduction

In many organizations, it is important that employees execute the tasks of processes in conformance with certain rules. For example, employees of a bank must check the credit history of a customer *before* granting a loan and call center agents must verify the identity of a caller *before* providing support. Possible implications of violating such rules might be severe. So-called *conformance checking* tools are able to automatically check whether the recorded process actions in an event log match the intended behavior as specified by a process model [1, 12, 14]. In this way, these tools intend to help organizations to automatically monitor the level of conformance and to identify problems.

However, automatically checking the conformance between event logs and process models is a complex task. The key challenge in this context is to create an *alignment* between the event log and the process model in order to explain the behavior as captured in the log using the process model. An approach commonly used to obtain such an alignment is the *cost-based approach*. This approach

aims to find the alignment with the least expensive deviations using a manually defined cost function [4]. One property of this cost-based approach is that it often leads to several alignments with the same costs. In practice, this means that an organization is provided with a set of possible root causes for non-conforming behavior, instead of with the most probable one. Recognizing this limitation, Alizadeh et al. [5, 6] introduced the notion of a *probable alignment*, which focuses on follow relationships in the fitted event log. However, their approach does not always return the most probable alignment and, therefore, also not the most probable root cause for non-conforming behavior.

In this paper, we propose a different technique for computing *probable alignments*. Our technique builds on the probabilities from the event log to compute the *most probable alignment*. More specifically, within the event log it assigns a probability to the occurrence of an activity; and in the transition system of the Petri net it assigns a probability to a transition. On this basis, we provide a principled approach that solves the most probable alignment problem in conformance checking.

The remainder of this paper is organized as follows. Section 2 discusses the background of our work. Section 3 introduces the preliminary concepts that we use. Section 4 explains our technique for computing probable alignments. Section 5 presents a theoretical evaluation of our technique. Section 6 discusses related work before Section 7 concludes the paper and discusses future work.

2 Background

In this section, we discuss the background of our work. Section 2.1 introduces the running example we use throughout this paper. Section 2.2 explains the details of the cost-based approach. Section 2.3 then discusses the shortcomings of current techniques for computing probable alignments.

2.1 Running Example

To illustrate the problem of finding the most probable alignment, consider the simple process model from Fig. 1, which captures the intended behavior of an organization as a Petri net. It defines that conforming behavior consists of a sequence of A, at least one B, followed by a choice between C and D. Thus, possible conforming traces to this model are: ⟨A,B,C⟩, ⟨A,B,B,C⟩, ..., ⟨A,B,D⟩, ⟨A,B,B,D⟩, and so on.

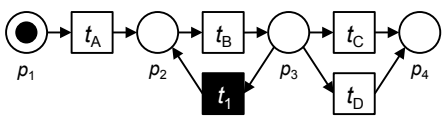


Fig. 1: Petri net capturing the intended behavior

Suppose the trace $\mathtt{tr}_1 = \langle$A,B,D,C$\rangle$ is observed, which is generated by an information system that tracks the actual behavior. This trace does not conform to the behavior specified by the process model from Fig. 1, since the process model defines a choice between C and D. The remainder of this section illustrates existing alignment techniques and their shortcomings based on this example.

2.2 Cost-Based Alignment

The cost-based alignment technique aligns each observed trace to a path in the process model; thereby it identifies missing events, additional events or incorrect ordering of events. Having modeled the trace $\mathtt{tr}_1$ as a sequential Petri net, the *Petri net product* in Fig. 2 models all possible movements by taking the product of the trace and process model. The Petri net of the trace represents log moves (additional events) and the Petri net of the process model represents model moves (missing events). Synchronous moves are created by pairing each activity in the trace to a transition in the model that corresponds to the same activity.

The transition system of the Petri net product is created, such that states are reachable markings and transitions are log, model and synchronous moves of an activity type. To find an optimal alignment, costs are assigned manually to the move types (i.e. log, model and synchronous move). The A^*-algorithm can then be used to find the shortest path from initial state to end state in the state space of the product net, which corresponds to a weighted transition system [1, 3, 4, 9].

For the trace $\langle$A,B,D,C$\rangle$, assuming equal costs for any deviation, the cost-based alignment finds two optimal alignments with the least expensive deviations (i.e. the alignments with the lowest costs), as displayed in Fig. 3. The $\gg$ symbol represents no progress in the replay on the respective side, e.g. the step for D in the alignment in Fig. 3a is a log move.

Note that the costs of a move type can be individually set for each activity. However, it is not possible to make the costs conditional on e.g. other activities in the sequence. This implies that move types of an activity are treated as independent from each other.

2.3 Shortcomings

Both optimal alignments in Fig. 3 have an equal cost of one log move, so either transition D or C is in excess. Without further knowledge, the choice between these alignments is arbitrary. This means that it is not guaranteed that the most probable root cause is taken.

Suppose in the event log $\langle$A,B,C$\rangle$ is observed 80 times and $\langle$A,B,D$\rangle$ is observed 20 times. Then $\langle$A,B,C$\rangle$ is the more likely alignment, so alignment 1 (see Fig. 3a) gets a higher probability than alignment 2 (see Fig. 3b).

The notion of a *probable alignment* as introduced by Alizadeh et al. [5] provides a solution by considering the event log history. They favor synchronous moves over log and model moves and they calculate a log move using the *never*

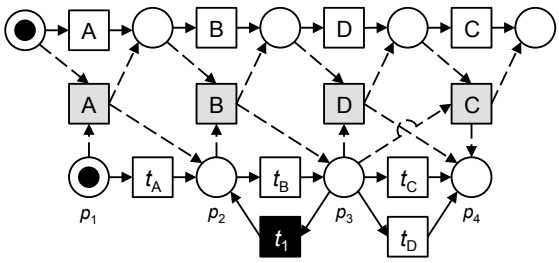


Fig. 2: Petri net product of the trace $\mathtt{tr}_1 = \langle$A,B,D,C$\rangle$ on top and the process model in Fig. 1 on the bottom. Synchronous execution is marked as shaded transitions in the middle.

trace	A	B	D	C
model	A	B	$\gg$	C
	t_A	t_B		t_C

(a) Optimal alignment 1

trace	A	B	D	C
model	A	B	D	$\gg$
	t_A	t_B	t_D	

(b) Optimal alignment 2

Fig. 3: Cost-based optimal alignments between the trace $\mathtt{tr}_1 = \langle$A,B,D,C$\rangle$ and the process model from Fig. 1

eventually follow relation conditioned on a conforming partial trace. For example, the probability that D never eventually follows after $\langle$A,B$\rangle$ (i.e. the log move probability of D given $\langle$A,B$\rangle$) is 80%. The log move probability of C given $\langle$A,B,D$\rangle$ is 100%. Maximizing the probable moves, the technique results in $\langle$A,B,D$\rangle$ (see Fig. 3b) as the most probable alignment, even though C occurred more frequently in the event log history.

In line with the intuition that C is more probable to occur, our approach identifies alignment 1 as the most probable alignment.

3 Preliminaries

This section gives a definition for *event log*, *Petri net* and *marking*.

An *event log* captures information about a running process, where each case (or instance) is represented with a trace of events that correspond to activities in the process. Formally, an event log is defined as follows.

Definition 1 (Event Log). *Let $\mathcal{A}$ be the set of activities, and $\mathcal{E}$ denote the set of events. The activity function $\alpha_L : \mathcal{E} \to \mathcal{A}$ assigns an activity to each event. A trace $\mathbf{tr}_E$ is a sequence of events, i.e., $\mathbf{tr}_E \in \mathcal{E}^*$. We use α_L to project sequences of events to activity traces $\mathbf{tr}_A \in \mathcal{A}^*$. Thus, an event log $\mathcal{L}$ is a multiset of activity traces.*

Table 1 captures the event log history used in the remainder of this paper. This history event log contains all events observed in the past and is non-empty.

Table 1: Event log history

trace	frequency
⟨A,B,C⟩	40
⟨A,B,D⟩	10
⟨A,B,B,C⟩	80
⟨A,B,B,D⟩	20
⟨A,B,B,B,C⟩	40
⟨A,B,B,B,D⟩	10
⟨A,B,B,B,B,B,B,C⟩	1
⟨A,B,B,B,B⟩	1
⟨A,C,B⟩	1

The corresponding process model is represented as a *Petri net* in Fig. 1. Note that the last two traces in Table 1 do not conform with the process model. Formally, a Petri net is defined as follows.

Definition 2 (Petri Net, Marking). *A* Petri net *over a set of activities $\mathcal{A}$ is a tuple $(P, \mathcal{T}, F, \alpha_1, m_i, m_f)$ where P and $\mathcal{T}$ are sets of places and transitions, respectively. $F : (P \times \mathcal{T}) \cup (\mathcal{T} \times P) \to \mathbb{N}$ is a flow relation between places and transitions; and $\alpha_1 : \mathcal{T} \nrightarrow \mathcal{A}$ is a partial function mapping transitions to activities. A state of a Petri net is determined by the* marking *($\mathcal{M} : P \to \mathbb{N}$) of a net, which specifies the number of tokens on the places. m_i and m_f are the initial marking and the final marking, respectively.*

In addition, a transition $t \in \mathcal{T}$ is *invisible* if $t \notin Dom(\alpha_1)$. A single activity can be represented by multiple transitions (i.e. *duplicate* transitions). Finally, we only consider *easy sound* Petri nets, that is, Petri nets of which the final state is reachable from its initial state [2].

4 Method

In this section, we introduce our method for computing the most probable alignment. Section 4.1 first introduces the probabilistic model. Section 4.2 then explains the computation of the most probable alignment.

4.1 Probabilistic Model

To derive the most probable alignment, we assess the probabilities of individual moves of one alignment. To this end, this section explains the probability calculation for a log-, model- and synchronous move of an activity.

Probability of an activity in the event log Given the event log history in Table 1, an activity X is a discrete random variable with an outcome in the set of activities $\mathcal{A} = \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}\}$. Let us extend the set of activities to allow for

unobserved activities $\mathcal{B}$, for example $\mathcal{B} = \{\star\}$. The probability of seeing outcome i is:

$$P(X = i) = \theta_i \; \forall \; i \in \mathcal{A} \cup \mathcal{B}, \tag{1}$$

where $\boldsymbol{\theta} = (\theta_{\mathrm{A}}, \theta_{\mathrm{B}}, \theta_{\mathrm{C}}, \theta_{\mathrm{D}}, \theta_{\star})$, $0 \leq \theta_i \leq 1$ and $\sum_{i \in \mathcal{A} \cup \mathcal{B}} \theta_i = 1$. So each (un)observed activity gets a (positive) probability of occurrence.

We assume that the random variables in the event log $X_1, \ldots, X_n$ are a *random sample* of size n from the population. This means that all random variables are independent and identically distributed (hereafter: iid). The sample is denoted by $\boldsymbol{X} = (X_1, \ldots, X_n)$ and is categorically distributed with parameter $\boldsymbol{\theta}$. We estimate $\boldsymbol{\theta}$ based on our event log history (see Table 1).

Let $\hat{\boldsymbol{\theta}}$ be the estimate of the true probability $\boldsymbol{\theta}$. Given our observations $\boldsymbol{X} = \boldsymbol{x}$, $\hat{\boldsymbol{\theta}}$ equals the sample mean:

$$\hat{\theta}_i = E(|X = i|) = \frac{|X = i|}{\sum_{j \in \mathcal{A} \cup \mathcal{B}} |X = j|} = \frac{|X = i|}{n} \; \forall \; i \in \mathcal{A} \cup \mathcal{B}, \tag{2}$$

So for our event log history in Table 1, the probability of seeing activity A ($\hat{\theta}_{\mathrm{A}}$) is $\frac{203}{817} = 24.9\%$, because Table 1 contains 203 A's of 817 activities in total. We do not observe activity $\star$, so $\hat{\theta}_{\star} = 0$.

To avoid zero probabilities, we can use Bayesian theory. In the Bayesian framework, the prior embodies a belief about the distribution of $\boldsymbol{\theta}$. Let parameter $\boldsymbol{\theta}$ be treated as a random variable: we define a prior and posterior density distribution of parameter $\boldsymbol{\theta}$. The posterior distribution is a combination of our prior belief and what we observe in the data. In determining the posterior distribution, the effect of the prior distribution decreases when the sample size increases [10]. For example, assuming each outcome in the event log is equally probable, the prior assigns $\frac{1}{5}$ to each outcome {A, B, C, D, $\star$}. Example 1 below illustrates the application of the prior[1].

Probability of a log move To calculate the probability of a log move, we want to remove the least likely fitting activity from the event log to align the remainder with the model. So the least frequent activity in an event log should be the most probable log move. To calculate the log move probability, we define the log move of an activity X^L as a monotone decreasing function of X ($X \mapsto h(X) = X^L$) [8], such that the estimated probability of a log move equals:

$$\hat{\theta}_i^L = E(|X^L = i|) = \frac{1 - \hat{\theta}_i}{k - 1} \; \forall \; i \in \mathcal{A} \cup \mathcal{B}, \tag{3}$$

[1] A conjugate prior means that the prior is from the same family of distributions as the posterior. The conjugate prior for the categorical distribution is the Dirichlet distribution [7].

i	$\lvert X = i \rvert$
A	1
B	1
C	1
D	1
$\star$	1
	5

(a) Assign prior

i	$\lvert X = i \rvert$
A	203+1
B	412+1
C	162+1
D	40+1
$\star$	1
	817+5

(b) Add observations

i	$\hat{\theta}_i$
A	$\frac{204}{822}$
B	$\frac{413}{822}$
C	$\frac{163}{822}$
D	$\frac{41}{822}$
$\star$	$\frac{1}{822}$
	1

(c) Calculate $\hat{\boldsymbol{\theta}}$

i	$\hat{\theta}_i^L$	
A	$\frac{618}{822 \cdot 4}$	≈ 0.19
B	$\frac{409}{822 \cdot 4}$	≈ 0.12
C	$\frac{659}{822 \cdot 4}$	≈ 0.20
D	$\frac{781}{822 \cdot 4}$	≈ 0.24
$\star$	$\frac{821}{822 \cdot 4}$	≈ 0.25
	1	

(d) Calculate $\hat{\boldsymbol{\theta}}^L$

Fig. 4: Estimation of the log move probability using Table 1

where k is the number of possible outcomes in $\mathcal{A} \cup \mathcal{B}$, $0 \leq \hat{\theta}_i^L \leq 1$, and $\sum_{i \in \mathcal{A} \cup \mathcal{B}} \hat{\theta}_i^L = 1$. It follows that log moves X^L are iid categorically($\hat{\boldsymbol{\theta}}^L$) distributed.

Example 1. Fig. 4 illustrates the estimation of the log move probability $\hat{\boldsymbol{\theta}}^L$. A prior is assigned to each activity $i \in \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}, \star\}$ in Fig. 4a, i.e. pseudo-observations are assigned to each activity. In Fig. 4b the observations of Table 1 are added to the prior. The probability of an activity $\hat{\boldsymbol{\theta}}$ is calculated with equation (2). The probability of a log move of an activity $\hat{\boldsymbol{\theta}}^L$ is calculated with equation (3), where $k = 5$, since i can take 5 different outcomes. Fig. 4c and 4d display the results. For example, the probability of a log move D equals 24%.

Probability of a model move The probability of a model move is equal to the probability of a transition given a marking in a Petri net[2]. Let marking M be the state of a Petri net. Given the Petri net in Fig. 1, a marking M is a discrete random variable with outcomes in the set of markings $\mathcal{M} = \{\{p_1\}, \{p_2\}, \{p_3\}, \{p_4\}\}$. A Petri net can be represented by a Markov Chain where states are markings, this means independence of firing history [13].

Given the Petri net in Fig. 1, a transition T is a discrete random variable with outcomes in the set of transitions $\mathcal{T} = \{t_\mathrm{A}, t_\mathrm{B}, t_\mathrm{C}, t_\mathrm{D}, t_1\}$. A transition conditioned on marking $M = m$ (hereafter: $T \mid m$) is also a discrete random variable. The probability of seeing outcome i given $M = m$ is:

$$P(T = i \mid M = m) = \phi_{i|m} \; \forall \; i \in \mathcal{T}, \tag{4}$$

where $0 \leq \phi_{i|m} \leq 1$ and $\sum_{i \in \mathcal{T}} \phi_{i|m} = 1$. $T \mid m$ is categorically distributed with parameter $\boldsymbol{\phi}_m = (\phi_{t_A|m}, \ldots, \phi_{t_1|m})$.

We estimate the true probability $\boldsymbol{\phi}_m$ by $\hat{\boldsymbol{\phi}}_m$ for all $m \in \mathcal{M}$. Given $\boldsymbol{T} = \boldsymbol{t} \mid \boldsymbol{M} = \boldsymbol{m}$, $\hat{\boldsymbol{\phi}}_m$ equals the sample mean:

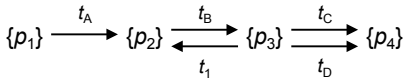
$$\hat{\phi}_{i|m} = E(|t = i \mid m|) = \frac{|t = i \mid m|}{\sum_{j \in \mathcal{T}} |t = j \mid m|} \; \forall \; t \in \mathcal{T} \tag{5}$$

[2] The probability of a transition given a marking uses the same logic as in Stochastic Petri Nets [13].

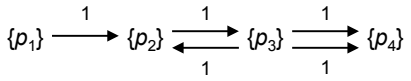
Moreover, we add a prior to embody our belief of the distribution of $\hat{\phi}_m$ for all $m \in \mathcal{M}$. The most frequent transition is the most probable model move, i.e. when an event is missing (a model move) the most likely activity is added. Model moves are independent, because markings M are memoryless.

Example 2. Fig. 5 illustrates the estimation of the model move probability $\hat{\phi}_m$. The states (markings) and the transitions of the Petri net are represented in a transition system in Fig. 5a. First, a prior is assigned to the transitions in Fig. 5b. Second, given the optimal *cost-based alignments*, the frequencies are calculated as the sum of synchronous and model moves for a transition. In Fig. 5c these frequencies are added to the prior. Third, this sum is used to estimate the probabilities with equation (5) in Fig. 5d.

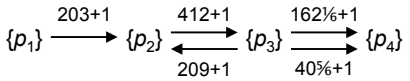
We use the cost-based alignment to calculate the observed random sample $\boldsymbol{T} = \boldsymbol{t} \mid \boldsymbol{M} = \boldsymbol{m}$. For this, we consider all optimal cost-based alignments with minimal number of model moves of invisible transitions. The cost-based alignment handles duplicate and invisible transitions in the model. Next to this, it non-deterministically chooses one alignment from all optimal alignments. In calculating our model move probabilities, we assume that all optimal alignments are equally likely, e.g. for $\texttt{tr}_1$ in Fig. 3 this implies that both alignments have a 50% likelihood (frequency = $\frac{1}{2}$).



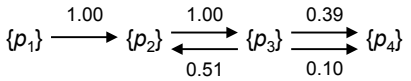
(a) The transition system of the process model in Fig. 1



(b) The transition system with assigned priors



(c) The transition system with assigned priors and aligned frequencies of synchronous and model moves



(d) The model move probabilities given a marking $\hat{\phi}_m$

Fig. 5: Estimation of the model move probability using the transition system of the process model in Fig. 1 and the optimal cost-based alignments

.

Probability of a synchronous move In the Petri net product of Fig. 2 we can perform either a synchronous move of A, or a model move t_{A} plus a log move of A. In line with existing techniques, we prefer a synchronous move.

Let ψ_i denote the probability of a synchronous move of activity $i \in \alpha_2(\mathcal{T})$, where $\alpha_2 : \mathcal{T} \nrightarrow \mathcal{A} \cup \mathcal{B}$ is a partial function mapping transitions to all activities $\mathcal{A} \cup \mathcal{B}$. We assume that the probability of the synchronous move is the maximum of the log and model move probability, so:

$$\psi_i = \max\{\theta_i^L, \phi_{i|m}\} \; \forall \; i \in \alpha_2(\mathcal{T}) \tag{6}$$

That is, the equation $\psi_i > \theta_i^L \ \phi_{i|m}$ is satisfied, which ensures that a synchronous move is preferred over a separate log plus model move.

Example 3. Given Fig. 4-5 and equation (6), the probability of a synchronous move of A equals $\psi_{\mathrm{A}} = \max\{\theta_{\mathrm{A}}^L, \phi_{t_{\mathrm{A}}|\{\mathrm{p}_1\}}\} = \max\{0.19, 1\} = 1$.

4.2 Most Probable Alignment

Consider the transition system of the Petri net product in Fig. 2, where we assign probabilities to the arcs. By the independence assumptions, the probabilities of successive moves (a path) can be multiplied. The most probable path is the path with the maximum probability.

A $\log(\cdot)$ transformation enables us to transform the product of probabilities to a sum of log probabilities. To find the most probable path, we search for the maximum sum. Maximizing the sum of $\log(\cdot)$ is equivalent to minimizing the sum of $-\log(\cdot)$. Hence, the $-\log(\cdot)$ transformation allows us to use the A^*-algorithm to find the shortest path in the transition system of the Petri net product of log and model. The shortest path with transformed probabilities then corresponds to the most probable alignment.

Example 4. Fig. 3 displays two alignments for the trace $\mathtt{tr}_1$. The probability for alignment 1 in Fig. 3a equals 9.3%:

$$\begin{aligned}\psi_{\mathrm{A}}\psi_{\mathrm{B}}\theta_{\mathrm{D}}^L\psi_{\mathrm{C}} &= \max\{0.19, 1\} \times \max\{0.12, 1\} \times 0.24 \times \max\{0.20, 0.39\} \\ &= 0.24 \times 0.39\end{aligned}$$

Similarly, the probability for alignment 2 in Fig. 3b equals 4.8%. Hence the most probable alignment for trace $\mathtt{tr}_1$ is alignment 1 (see Fig. 3a).

5 Theoretical Evaluation

This section theoretically evaluates the performance of our approach.

Given the process model in Fig. 1 and the event log in Table 1 we calculate the most probable alignment for four traces: $\mathtt{tr}_1 = \langle$A,B,D,C$\rangle$, $\mathtt{tr}_2 = \langle$A,B,C,D$\rangle$, $\mathtt{tr}_3 = \langle$A,B,B,B,B$\rangle$ and $\mathtt{tr}_4 = \langle$A,A,B,C$\rangle$. Two techniques are applied; the technique from Alizadeh et al. [5] (see Fig. 6)[3] and our technique (see Fig. 7). In this section, we show that our technique improves upon the shortcomings of [5].

First, the order of transitions in the trace (log moves) determines the outcome (see Fig. 6a and 6c and Section 2). Given that the number of observations of C

[3] To calculate probabilities Alizadeh et al. [5] use the fitted event log ($\mathcal{L}_{\mathrm{fit}}$) of Table 1 (first 6 rows). Process executions are mapped onto a state, for which a state representation function is used: either a sequence, multiset or set abstraction. As a cost function $g = 1 + \log\left(\frac{1}{\theta}\right)$ is used.

is larger than the number of observations of D in Table 1, a synchronous move of C is more probable than a synchronous move of D independent of the order, as is shown in Fig. 7a and 7c.

Second, the model behavior is restricted to behavior in the fitting part of the log $\mathcal{L}_{\text{fit}}$ (see Fig. 6b). $\mathcal{L}_{\text{fit}}$ contains traces with 1,2,3 and 7 repetitions of activity B. For trace $\mathtt{tr}_3$, Alizadeh et al. [5] find the outcome ⟨A,B,B,B,C⟩ if the state representation function is a sequence or multiset abstraction[4]. We argue that it is more likely that the 4th B is a synchronous move instead of a log move, given the Markov property of the process model. We thus get the outcome ⟨A,B,B,B,B,C⟩, as is shown in Fig. 7b.

Third - in line with [4] - the location of a synchronous move does not matter in a repetition of log moves, where one synchronous move is possible according to the model. So we find two alignments in Fig. 7d. In contrast, using [5] the synchronous move is assigned to the first activity of the repetition (see Fig. 6d).

trace	A	B	D	C
model	A	B	D	≫
	t_A	t_B	t_D	

(a) $\mathtt{tr}_1 = \langle$A,B,D,C$\rangle$

trace	A	B	≫	B	≫	B	B	≫
model	A	B	τ	B	τ	B	≫	C
	t_A	t_B	t_1	t_B	t_1	t_B		t_C

(b) $\mathtt{tr}_3 = \langle$A,B,B,B,B$\rangle$ and sequence or multiset abstraction

trace	A	B	C	D
model	A	B	C	≫
	t_A	t_B	t_C	

(c) $\mathtt{tr}_2 = \langle$A,B,C,D$\rangle$

trace	A	A	B	C
model	A	≫	B	C
	t_A		t_B	t_C

(d) $\mathtt{tr}_4 = \langle$A,A,B,C$\rangle$

Fig. 6: The probable alignments for traces $\mathtt{tr}_1, \mathtt{tr}_2, \mathtt{tr}_3$ and $\mathtt{tr}_4$ with cost function $g = 1 + \log(\frac{1}{\theta})$ according to [5]

trace	A	B	D	C
model	A	B	≫	C
	t_A	t_B		t_C

(a) $\mathtt{tr}_1 = \langle$A,B,D,C$\rangle$

trace	A	B	≫	B	≫	B	≫	B	≫
model	A	B	τ	B	τ	B	τ	B	C
	t_A	t_B	t_1	t_B	t_1	t_B	t_1	t_B	t_C

(b) $\mathtt{tr}_3 = \langle$A,B,B,B,B$\rangle$

trace	A	B	C	D
model	A	B	C	≫
	t_A	t_B	t_C	

(c) $\mathtt{tr}_2 = \langle$A,B,C,D$\rangle$

trace	A	A	B	C
model	A	≫	B	C
	t_A		t_B	t_C

and

trace	A	A	B	C
model	≫	A	B	C
		t_A	t_B	t_C

(d) $\mathtt{tr}_4 = \langle$A,A,B,C$\rangle$

Fig. 7: The most probable alignments for traces $\mathtt{tr}_1, \mathtt{tr}_2, \mathtt{tr}_3$ and $\mathtt{tr}_4$ using our technique

[4] Alizadeh et al. [5] find the outcome ⟨A,B,B,B,B,C⟩ if the state representation function is a set abstraction. Our technique obtains the same result (see Fig. 7b).

Beyond addressing the shortcomings of [5], our approach to determine the most probable alignment is unaffected by noise. This is another strength. We consider two extreme variants of noise: (1) An activity is not in the model, but is in the log. (2) An activity is not in the log, but is in the model. The prior assigns a positive probability to the model and log moves (and therefore the synchronous move) of that activity. Neither of these has an effect on the probability estimation[5].

We always prefer synchronous moves, even if frequencies in the event log are missing or not balanced. Suppose in the event log of the running example (see Fig. 1) activity C is observed 999 times and D is observed once; and we observe a trace $\mathtt{tr}_5 = \langle \text{A,B,D} \rangle$. A synchronous move of D is preferred over a log move of D plus a model move of C (due to equation (6)).

6 Related Work

The cost-based alignment technique is used to calculate the fitness conformance metric. Fitness measures how well the process model captures the observed behavior as recorded in an event log.

In this paper we focus on the control-flow perspective. Some approaches extend alignment-based techniques to support conformance checking based on multiple perspectives (e.g. control-flow, data, resource and time). De Leoni and Van der Aalst [11] build alignments by first considering the control-flow and, second, refine the computed alignments using other perspectives. In contrast, Mannhardt et al. [12] balance the deviations with respect to all perspectives (not prioritizing control-flow). Both approaches use a (customizable) cost function and do not consider a cost function based on probabilities. Alizadeh et al. [6] consider multiple perspectives to calculate probable alignments, but their technique has the same shortcoming as [5] (mentioned in Sections 2.3 and 5).

7 Conclusion

In this paper, we address the problem of computing the *most probable alignment* in the context of conformance checking. The core idea of our approach is to use the history of the event log to assign probabilities to the occurrence of activities and to the transitions between them. We apply Bayesian theory to avoid zero probabilities. The theoretical evaluation demonstrates that our approach improves upon existing work by Alizadeh et al. [5]. Moreover, it is unaffected by noise.

In future work, we plan to extend this current work with an empirical evaluation. To this end, we intend to implement the presented technique in the context of the ProM Framework. Further, we plan to relax the independence assumption of activities in the event log to take into account the correlation between activities, e.g. long distance dependencies. Finally, we intend to extend our approach

[5] Note for both variants $\mathcal{L}_{\text{fit}} = \emptyset$, so the technique in [5] does not work.

to the multi-perspective scenario, in which we would take into account attribute values.

Acknowledgement We thank Massimiliano de Leoni for validating our understanding of [5].

References

[1] van der Aalst, W. M. P., Adriansyah, A., and van Dongen, B. F. (2012). Replaying history on process models for conformance checking and performance analysis. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2:182–192.

[2] van der Aalst, W. M. P., van Hee, K. M., ter Hofstede, K. M., Sidorova, N., Verbeek, H. M. W., Voorhoeve, M., and Wynn, M. T. (2011). Soundness of workflow nets: classification, decidability, and analysis. *Formal Aspects of Computing*, 23:333–363.

[3] Adriansyah, A., van Dongen, B. F., and van der Aalst, W. M. P. (2011). Conformance checking using cost-based fitness analysis. In *EDOC, 2011 15th IEEE International*, pages 55–64.

[4] Adriansyah, A., van Dongen, B. F., and van der Aalst, W. M. P. (2013). Memory-efficient alignment of observed and modeled behavior. In *BPM Center Report*. 03-03.

[5] Alizadeh, M., de Leoni, M., and Zannone, N. (2014). History-based construction of log-process alignments for conformance checking: Discovering what really went wrong. *SIMPDA*, pages 1–15.

[6] Alizadeh, M., de Leoni, M., and Zannone, N. (2015). Constructing probable explanations of nonconformity: A data-aware and history-based approach. In *Computational Intelligence, 2015 IEEE Symposium Series on. IEEE*.

[7] Barber, D. (2012). *Bayesian reasoning and machine learning*. Cambridge University Press.

[8] Casella, G. and Berger, R. L. (2002). *Statistical inferences*, volume 2. Pacific Grove, CA: Duxbury.

[9] Dechter, R. and Pearl, J. (1985). Generalized best-first search strategies and the optimality of A. *Journal of the ACM (JACM)*, 32:505–536.

[10] Greenberg, E. (2012). *Introduction to Bayesian econometrics*. Cambridge University Press.

[11] de Leoni, M. and van der Aalst, W. M. P. (2013). Aligning event logs and process models for multi-perspective conformance checking: An approach based on integer linear programming. *BPM*, pages 113–129.

[12] Mannhardt, F., de Leoni, M., Reijers, H. A., and van der Aalst, W. M. P. (2016). Balanced multi-perspective checking of process conformance. *Computing*, 98:407–437.

[13] Molloy, M. K. (1982). Performance analysis using stochastic Petri nets. *IEEE Transactions on computers*, 31:913–917.

[14] Munoz-Gama, J., Carmona, J., and van der Aalst, W. M. P. (2014). Single-entry single-exit decomposed conformance checking. *Information Systems*, 46:102–122.