

Received October 4, 2021, accepted November 7, 2021, date of publication November 16, 2021, date of current version December 10, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3128571

An Approach for Mining Multiple Types of Silent Transitions in Business Process

LI-LI WANG¹, XIAN-WEN FANG¹, CHI-FENG SHAO², AND ESTHER ASARE¹

¹College of Mathematics and Big Data, Anhui University of Science and Technology, Huainan 232000, China

²College of Information and Network Engineering, Anhui Science and Technology University, Bengbu 233100, China

Corresponding author: Xian-Wen Fang (xwfang@aust.edu.cn)

This work was supported in part by the National Natural Science Foundation, China, under Grant 61572035 and Grant 61402011; in part by the Natural Science Foundation of Anhui Province, China, under Grant 2008085QD178; and in part by the Anhui Province College Excellent Young Talents Fund Project of China under Grant gxyqZD2020020.

ABSTRACT The purpose of process discovery is to construct a process model based on business process execution data recorded in an event log. Many situations may lead to silent transitions that appeared in the process model, while the execution of silent transitions is not recorded in event logs. Therefore, mining silent transitions has been one of the difficult problems in process mining. Existing approaches have some limitations on discovering the silent transition in concurrent structures and may produce many redundant silent transitions which make discovered process model complicated. A novel approach to discover multiple types of silent transitions from an event log is presented in the paper. The basic behavior relationship between activity pairs based on the event log is used to construct the process model with silent transitions of and-gateway type and loop type. Meanwhile, the technique of behavior distance is proposed to discover silent transitions of skip type. Finally, the process model with multiple types of silent transitions is obtained. Experimental results show that the proposed approach can find multiple types of silent transitions correctly, and the number of redundant silent transitions is much less than the existing methods. Meanwhile, it significantly improves the F-measure of the model.

INDEX TERMS Behavior distance, labeled Petri net, process discovery, silent transition.

I. INTRODUCTION

Nowadays, information system records the real execution of the business process in the form of event logs. The goal of process mining is to discover, monitor and enhance real life processes by automatically extracting valuable information from these event logs. It includes three main domains, i.e., process discovery, conformance checking and process enhancement. However, process discovery is the necessary technique for the other domains, which automatically constructs process models by applying the different discovery techniques on event logs [1]. Until recently, dozens of process discovery algorithms have been proposed [2]–[4]. Whereas, many problems need to be solved in process mining, such as short loops, duplicated transitions, invisible (or silent) transitions, non-free-choice constructs, noises, infrequent behaviors, incompleteness, etc [5]. In addition, privacy-preservation issues of cross-organization business process mining [6], [7],

efficiency problem when dealing with large-scale event logs [8], process model discovery and process similarity measure considering both control-flow structures and data-flow information [9]. However, in this paper, we focus on the detection of silent transitions from event logs. Various factors in real life may cause silent transitions to appear in the process model, and these silent transitions only play a role of routing but cannot be ignored. Silent transitions are difficult to discover because they do not appear in any event trace. The main reasons for leading to silent transitions are as follows [10]:

- There are some actual tasks that can be allowed to be missing in some event traces.
- There are meaningless tasks only used for routing purposes in process models.
- The enactment services of process models allow skipping or redoing the current task and jumping to any previous task. But such execution logic is not expressed in the control logic of the process model.

The associate editor coordinating the review of this manuscript and approving it for publication was Yue Zhang¹.

Until now, there are several mining approaches that are capable of discovering silent transitions [10]–[19], such as α # algorithm [10], [11], genetic algorithm [12], α \$ [15], Inductive Mining (IM) algorithm [2], Coupled Silent Markov Model-Nonfree Choice Invisible Task (CHMM_NCIT) [17], etc. These mining algorithms use techniques such as directly-follows relationships, Markov model, or genetic algorithm to detect silent tasks. However, some yield a lot of redundant silent tasks, and some fail to detect silent tasks that should be appeared in the concurrent structure. Thus, the existing process discovery algorithms have difficulties to discover possible silent tasks from the given event logs accurately and guarantee to return the appropriate result. They pay less attention to the behavioral relationship between activities in the event logs when detecting silent transitions. Thus, a novel approach to discover silent transitions based on the behavior distance is presented in the paper. The proposed approach can discover multiple types of silent transitions and deal with silent transitions that appeared in concurrent structures. Meanwhile, it will not produce a large number of redundant silent transitions. Results of experiments on event logs show that the proposed method is promising compared to existing methods. The main works of the paper are as follows:

- 1) We present a new technique of behavior distance which can quantify the behavior relationship between activities in the event logs. Subsequently, the behavior distance between activities based on log, model, and concurrent structure is presented respectively to discover silent transitions accurately.
- 2) The traditional behavioral profile is refined to more accurately capture the behavior relationship between activities. We construct the process model with silent transitions by analyzing the behavior relationship between activities in the event logs.
- 3) The novel approach to discover multiple silent transitions is proposed based on behavior distances and basic behavioral relations of event logs. The proposed approach can find multiple types of silent transitions correctly, and the number of redundant silent transitions is much less than the existing methods.

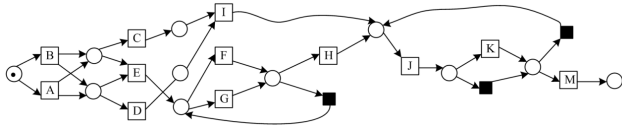
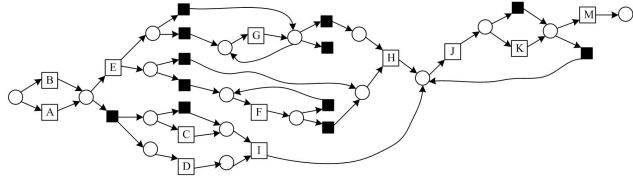
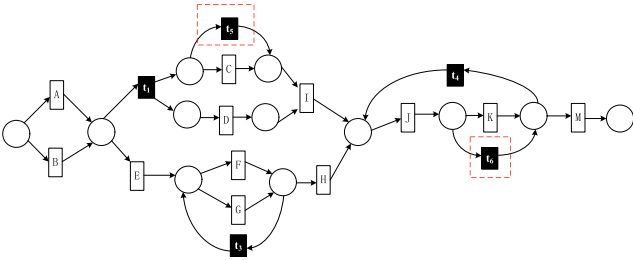
The remainder of the paper is structured as follows. Section II reviews works on the silent task. Section III introduces a motivation example. Section IV presents the preliminary concepts. The approach for discovering silent transitions of and-gateway type and loop type is given in Section V. Further on, the technology of discovering silent transitions of skip type is described in Section VI. Section VII shows the evaluation results of the proposed approach. Finally, Section VIII concludes the paper and discusses future work.

II. RELATED WORKS

Over the last decade, there are several mining techniques which support the detection of silent transitions, such as IM algorithm [2], α # algorithm [10], [11], genetic algorithm [12], α \$ [15], Coupled Silent Markov Model-Invisible

Task (CHMM-IT) [16], Coupled Silent Markov Model-Nonfree Choice Invisible Task (CHMM_NCIT) [17], etc. The IM algorithm [2] obtains a process model with silent transitions using the technique of process tree cutting. Still, it may yield too many redundant silent transitions, which makes the process model very complicated. The literature [11] gives a mining method of a process model with prime invisible tasks based on the literature [10]. This approach detects invisible tasks by identifying the mendacious dependency, which is separated from the causal dependency. Therefore, if the tasks have no directly-follows relation in the event log, then the silent task between them will not be detected. Thus, it is difficult to detect the silent transition appeared in concurrent structure. In [12], the authors propose the genetic mining algorithm, which supports the detection of invisible tasks, duplicated tasks and non-free choice constructs. Whereas this algorithm needs many user-defined parameters and it cannot always guarantee to return the most appropriate results. A Synchro-net based mining algorithm, which is a synchronization-based model of workflow logic and workflow semantics is proposed in [13]. The authors state that short-loops and invisible tasks can be dealt with at ease. However, neither the original model nor the mined model contains any invisible task. Literature [14] presents a decisions mining approach to mine decisions from process logs, which emphasizes detecting data dependencies that affect the routings of cases. When interpreting the control-flow semantics at a decision point, the authors propose an approach to identify decision branches starting from invisible tasks. However, not all kinds of silent tasks can be dealt with by this approach. The literature [15] provides the α \$ algorithm to handle silent transitions in non-free-choice Petri net, but it still cannot detect the silent transition that appeared in concurrent structure. Literatures [16], [17] present a method that utilizes silent Markov to construct process models with a free and non-free choice construct containing invisible prime tasks from incomplete event log, both Coupled Silent Markov Model-Nonfree Choice Invisible Task (CHMM_NCIT) and Coupled Silent Markov Model-Invisible Task (CHMM-IT) use Coupled Silent Markov Model (CHMM) and use Baum-Welch algorithm to determine the weights of variables of CHMM. However, the time complexity of these algorithms is very high. In [18], the authors propose a method to construct silent task and silent task in non-free choice relation when converting a declarative model into an imperative model in the form of Petri Net model. Therefore, this method cannot discover silent task when constructing the process model from the event logs. In literature [19], a graph-based algorithm is proposed to discover silent transitions. However, when event logs are very large, it is difficult to get database graphs to express the dependency between events.

All methods mentioned above, none of them concentrate on the behavioral relation between activities when detecting the silent task, and they still cannot handle all kinds of silent transitions well without too many redundant silent transitions. This paper presents a novel approach to discover silent

FIGURE 1. A process model M_1 obtained by $\alpha \#$ algorithm.FIGURE 2. A process model M_1' obtained by IM algorithm.FIGURE 3. A process model M_1'' obtained by the proposed approach.

transitions based on behavioral relationship without generating a large number of redundant silent transitions.

III. MOTIVATION

In the following, we will illustrate the limitation of existing approach to mine silent transitions with an example. For the event log $L_1 = \{\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6, \sigma_7, \sigma_8, \sigma_9, \sigma_{10}, \sigma_{11}, \sigma_{12}, \sigma_{13}\}$ where, $\sigma_1 = ACDIJKM$, $\sigma_2 = ADCIJKJKJ KM$, $\sigma_3 = ADIJKM$, $\sigma_4 = BDCIJKJKM$, $\sigma_5 = BCDIJJJM$, $\sigma_6 = BDIJKJJKM$, $\sigma_7 = ADIJKJKJJKJKM$, $\sigma_8 = AEFHJKJKM$, $\sigma_9 = AEGHJKM$, $\sigma_{10} = BEGGHJM$, $\sigma_{11} = BEFFHJKM$, $\sigma_{12} = BEGFGFFGGHJMM$, $\sigma_{13} = AEGGGFGFFHJKM$. The frequency of traces is not considered in the paper. The process models shown in Figure 1 and Figure 2 are obtained by applying the approach proposed in the literature [3] and literature [12] on the above log respectively. Applying the proposed method in this paper, the process model can be yielded shown in Figure 3.

The literature [3] has difficulties to discover the silent transition between activities pairs that have no directly-follows relation, such as the silent transition t_5 shown in Figure 3. The Inductive Mining method [12] produces lots of redundant silent transitions, which lead to the model complex, incomprehensible and inaccurate analysis results. To address the aforementioned problems, a novel method of discovering a process model with non-redundant silent transitions is presented in this paper, which overcomes the drawbacks of existing approaches to mining silent transitions. Meanwhile,

the effectiveness of the method through a set of experiments is validated.

IV. PRELIMINARIES

The basic definitions and related knowledge of Petri net will not be introduced in this paper. More details can be found in the literature [20]. The basic knowledge of process mining can be found in the literature [1]. Here, we only present basic preliminaries and notations, used throughout the paper.

Definition 1 ([21] Labeled Petri Net): A labeled Petri net is a 4-tuple $N := (P, T, F, \lambda)$, where (P, T, F) is a Petri net, \mathcal{A} represents the universe of labels to describe actions that have explicit meaning, τ represents the particular label that has no domain interpretation, where $\tau \notin \mathcal{A}$. $\lambda : T \rightarrow \mathcal{A} \cup \{\tau\}$ is a function that assigns labels to transitions. If $\lambda(t) \neq \tau$, then t is observable. Otherwise, t is silent or invisible.

A labeled net system $S := (P, T; F, \lambda, M_0)$ is a labeled net $N := (P, T; F, \lambda)$ with initial marking M_0 . The semantics of a labeled net is the same as for an unlabeled net. For each occurrence sequence $\sigma = t_1 t_2 t_3 \dots t_k$ of S induces the labeled occurrence sequence $l(\sigma) = l(t_1)l(t_2)l(t_3) \dots l(t_k)|_{\mathcal{A} \setminus \{\tau\}}$ obtained by replacing each transition t_i by its label $l(t_i)$ and omitting all τ from the result by projection onto $\mathcal{A} \setminus \{\tau\}$. We say that N can replay a log L iff each $\sigma \in L$ is a labeled occurrence sequence of N .

Definition 2 ([21] Labeled Execution Sequence or Labeled Occurrence Sequence): Let $S := (N, M_0)$ be a labeled net system, where $N := (P, T, F, \lambda)$. All labeled execution sequences of a net system S are all lists of the form $\{a_i\} \cdot (\mathcal{A} \setminus \{\tau\})^* \cdot \{a_f\}$ that follow the executable semantic of S , where a_i, a_f is an artificial start transition and end transition respectively. All labeled execution sequences of S are denoted as $\mathcal{T}(S)$.

Definition 3 ([22] Directly-Follows Relation (Log)): Let L be an event log over T . For $a, b \in T$, if existing a trace $\sigma \in L$ and $\sigma = t_1 t_2 \dots t_n$ such that $t_i = a, t_{i+1} = b$ (where $i \in \{1, 2, \dots, n-1\}$), we say that there is a directly-follows relation between a and b , which is written as $a >_L b$.

Definition 4 ([22] Weak Order (Log)): Let L be an event log over T . For $a, b \in T$, if existing a trace $\sigma \in L$ and $\sigma = t_1 t_2 \dots t_n$ such that $t_i = a, t_j = b$ (where $1 \leq i < j \leq n$), we say that there is a weak order relationship between a and b , which is written as $a \geq_L b$.

Definition 5 ([1] Free-Choice): A Petri net is free-choice if any two transitions sharing an input place have identical input sets, i.e., $\bullet t_1 \cap \bullet t_2 = \emptyset$ implies $\bullet t_1 = \bullet t_2$ for any $t_1, t_2 \in T$.

A Petri net is sound, if it is safeness, proper completion and has no dead transitions, meanwhile, all transitions are on the path from input place i and output place o .

V. CONSTRUCTING A PROCESS MODEL WITH SILENT TRANSITIONS OF AND-GATEWAY AND LOOP TYPE

On the basis of literature [23], this section further points out how to associate the behavioral characteristic relationship between activities in the event log and their structural

relationship in the process model through the behavioral relationship and behavioral distance in the log. The correctness of the corresponding relation is proved by propositions, which provides a theoretical basis for finding the initial process model with silence transition of and-gateway type and loop type from event log. Moreover, the approach to discover silent transitions of and-gateway and loop type from event log is provided.

Obviously, directly-follows relation and weak order relation between activity pairs are qualitative analysis methods to behavior relation. However, behavioral distance is presented in the following, which realizes the quantitative analysis of behavioral relations

Definition 6 (Trace-Based Behavior Distance): Let L be an event log over T . For $a, b \in T$, $\sigma \in L$, the behavior distance between a and b is defined as:

$$BDis(a, b, \sigma) = \begin{cases} \{k | 1 \leq i \leq |\sigma| \wedge [\sigma(i) = a \wedge \sigma(i+k) = b]\} & \text{if } a \succeq_L b \\ -BDis(b, a, \sigma) & \text{if } b \succeq_L a \wedge a \not\prec_L b \end{cases} \quad (1)$$

Definition 7 (Log-Based Maximum Behavior Distance and Minimum Behavior Distance): Let L be an event log over T . For $a, b \in T$, log-based minimum behavior distance $BDis_{\min}(a, b, L)$ and log-based maximum behavior distance $BDis_{\max}(a, b, L)$ are defined as:

$$BDis_{\min}(a, b, L) = k \Leftrightarrow \nexists \sigma \in L, \quad BDis(a, b, \sigma) < k, \quad (2)$$

$$BDis_{\max}(a, b, L) = k' \Leftrightarrow \nexists \sigma \in L, \quad BDis(a, b, \sigma) > k' \quad (3)$$

If an activity a and an activity b are in an exclusive relationship, i.e., they can never both appear in a trace, then $BDis_{\min}(a, b, L) = BDis_{\max}(a, b, L) = \infty$. If $a \succeq_L b$ and $b \not\prec_L a$ holds in any trace, then $BDis_{\min}(b, a, L) = -BDis_{\min}(a, b, L)$ and $BDis_{\max}(b, a, L) = -BDis_{\max}(a, b, L)$.

Given an event log L , A_L represents a set of activities appeared in the log L . Let $|A_L| = n$, $BDisMatrix$ is a $n \times n$ dimensional matrix which denotes the behavior distance between all activities based on the event log. For $\forall i, j \in \{1, \dots, n\}$, let the i -th row corresponds to activity a and the j -th column corresponds to activity b , the value of the i -th row and j -th column of the matrix denoted as $M_{i,j} = [x, y]$, where $x = BDis_{\min}(a, b, L)$, $y = BDis_{\max}(a, b, L)$.

The behavior distance matrix of the event log L_1 shown in Section II is in Figure 4.

Literature [11] indicates that the mendacious dependencies are important for the discovery of silent transitions. However, classical behavioral profiles do not distinguish mendacious dependency from directly-follows dependency. As we know that the cyclic structures have a substantial impact on the interleaving behavioral relations, for example, two exclusive transitions inside a cycle. Therefore, it cannot determine

Activity	A	B	C	D	E	F	G	H	I	J	K	M
A	∞	∞	1,2	1,2	1,1	2,5	2,2	3,9	2,3	3,10	4,11	5,12
B	∞	∞	1,2	1,2	1,1	2,3	2,4	5,8	2,3	3,9	4,7	7,11
C			∞	1,1	∞	∞	∞	∞	1,2	2,3	3,4	5,8
D				∞	∞	∞	∞	∞	1,2	2,3	3,4	4,12
E					∞	1,4	1,3	2,8	∞	3,9	4,10	5,11
F						1,2	1,1	1,3	∞	2,4	3,3	4,6
G							1,3	1,3	∞	2,4	3,5	4,6
H								∞	∞	1,1	2,2	3,5
I									∞	1,1	2,2	3,11
J										1,2	1,1	1,2
K											2,4	1,1
M												∞

FIGURE 4. The behavior distance matrix of L_1 .

Activity	A	B	C	D	E	F	G	H	I	J	K	M
A	#	#	\rightarrow	\rightarrow	\rightarrow	\rightarrow	\rightarrow	\rightarrow	\rightarrow	\rightarrow	\rightarrow	\rightarrow
B	#	#	\rightarrow	\rightarrow	\rightarrow	\rightarrow	\rightarrow	\rightarrow	\rightarrow	\rightarrow	\rightarrow	\rightarrow
C			#		#	#	#	#	\rightarrow	\rightarrow	\rightarrow	\rightarrow
D				#	#	#	#	#	\rightarrow	\rightarrow	\rightarrow	\rightarrow
E					#	\rightarrow	\rightarrow	\rightarrow	\rightarrow	\rightarrow	\rightarrow	\rightarrow
F								\rightarrow	#	\rightarrow	\rightarrow	\rightarrow
G								\rightarrow	#	\rightarrow	\rightarrow	\rightarrow
H								#	\rightarrow	\rightarrow	\rightarrow	\rightarrow
I									#		\rightarrow	\rightarrow
J										'	'	\rightarrow
K											'	\rightarrow
M												#

FIGURE 5. Basic behavior relationship between activities based on L_1 .

the structural characteristics between activities, such as true concurrency and cyclic structure, when activity pairs are in interleaving behavioral relations. Obviously, existing behavioral profile is too rough. It is very important to identify causal dependency, mendacious dependency and different interleaving order relationships for the detection of silent transition. Therefore, Definition 8 presents basic behavior relation based on log.

Definition 8 (Basic Behavior Relation Based on Log): Let L be an event log over T , let $a, b \in T$,

- The causal relation \rightarrow_L , iff $a \succ_L b$, $b \not\prec_L a$ and $BDis_{\min}(a, b, L) = BDis_{\max}(a, b, L) = 1$.
- The strict order relation \rightarrow_L , iff $a \succeq_L b$, $b \not\prec_L a$ and $BDis_{\max}(a, b, L) > 1$.
- The exclusiveness relation $+_L$, iff $a \not\prec_L b$ and $b \not\prec_L a$.
- The concurrent interleaving order relation $||_L$, iff $a \succeq_L b$, $b \succeq_L a$, $b \not\prec_L b$ and $a \not\prec_L a$.
- The loop interleaving order relation $||'_L$, iff $a \succeq_L b$, $b \succeq_L a$, $b \succeq_L b$ and $a \succeq_L a$.

The set $\rightarrow_L, \rightarrow_L, +_L, ||_L, ||'_L$ is the basic behavior relation of L .

For the event log shown in Section II, the basic behavior relationship between activities is shown in Figure 5.

Definition 9 presents the definition of various types of silent transition according to their structural character.

Definition 9 (Silent Transition Type): Let $N := (P, T, F, \lambda)$ be a labeled Petri net, $\mathcal{A} \cup \tau$ represent all labels of activities. For $t_1, t_2, t \in \mathcal{A} \cup \tau$, if

- (1) $t_1 \rightarrow_N t_2$, $t_1 \bullet \cap \bullet t \neq \emptyset$ and $t \bullet \cap \bullet t_2 \neq \emptyset$, where $\lambda(t_1) \neq \tau$, $\lambda(t_2) \neq \tau$, $\lambda(t) = \tau$ holds, then t is called as a silent transition of skip type.
- (2) $t_1 \rightarrow_N t_2$, $t_1 \bullet \cap \bullet t \neq \emptyset$ and $\bullet t \cap \bullet t_2 \neq \emptyset$, where $\lambda(t_1) \neq \tau$, $\lambda(t_2) \neq \tau$, $\lambda(t) = \tau$ holds, then t is called as a silent transition of loop type.

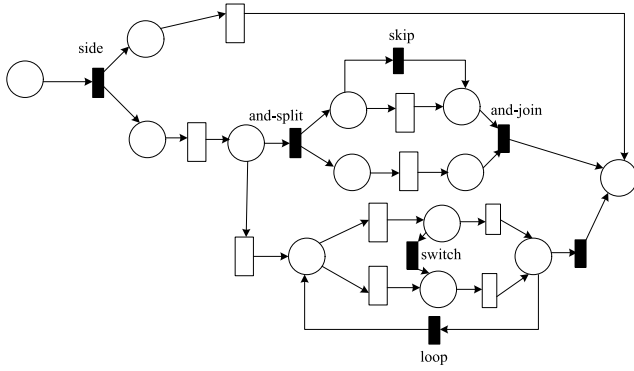


FIGURE 6. Example of silent transitions.

- (3) $t_1 \#_N t_2$, $t_1^\bullet = {}^\bullet t$ and $t^\bullet = {}^\bullet t_2$, where $\lambda(t_1) \neq \tau$, $\lambda(t_2) \neq \tau$, $\lambda(t) = \tau$ holds, then t is called as a silent transition of switch type.
- (4) $\lambda(t_1) = \lambda(t_2) = \tau$, $t_1^\bullet = 1$, $t_1^\bullet > 1$ or $t_2^\bullet > 1$, $t_2^\bullet = 1$ holds, then t_1, t_2 is called as a silent transition of and-split type and and-join type respectively. Both of them are called as silent transitions of and-gateway type.
- (5) $\lambda(t) = \tau$, $t^\bullet = \{s\}$, $s = \emptyset$, $|t^\bullet| > 1$ or $t^\bullet = \{s\}$, $s = \emptyset$, $|t^\bullet| > 1$ holds, then t is called as a silent transition of side type.

Figure 6 illustrates silent transitions of side type, skip type, loop type, switch type, and and-gateway type.

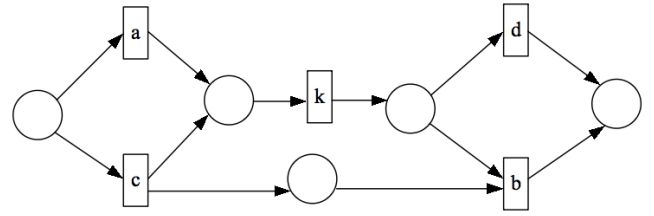
Propositions 1 to 5 demonstrate the relations between based-log the basic behavior relations of activities and their structure in a sound free-choice Petri net. These propositions provide a modeling basis for constructing business process models from the event log.

Proposition 1: If activity a and activity b are in causal dependence based on log, then they are in structural sequence relation in the Petri net and $a^\bullet = {}^\bullet b$, $|a^\bullet| = |{}^\bullet b| = 1$ hold.

Proof: We prove it in three cases. Case 1: Assume $a^\bullet \neq {}^\bullet b$ and $|a^\bullet| = |{}^\bullet b| = 1$. Obviously, this leads to a contradiction with $a \rightarrow_L b$. Case2: Assume $a^\bullet = {}^\bullet b$ and $|a^\bullet| = |{}^\bullet b| \neq 1$. This indicates that there are multiple connection places between activity a and activity b . Only preserving one place by deleting others places does not affect the behavioral relationships between activity a and activity b . Case 3: Assume $a^\bullet \neq {}^\bullet b$ and $|a^\bullet| \neq |{}^\bullet b|$. Since $a \rightarrow_L b$, we can infer $a^\bullet \cap {}^\bullet b \neq \emptyset$. Without loss of generality, we just discuss the simplest case. Let $a^\bullet \cap {}^\bullet b = p$, we assume $|a^\bullet| = 2$, $|{}^\bullet b| = 1$, let $a^\bullet = \{p, p'\}$, $(p')^\bullet = c$. When the activity a is fired, activity b and activity c are in concurrent. Hence, there exists an execution sequence in the form of $\dots acb \dots$, which produces $BD_{\max}(a, b) > 1$. Therefore, this leads to a contradiction. \square

Proposition 2: If activity a and activity b are in strict order relation based on log, then there exists a path composed of transitions and places between activity a and activity b in a Petri net.

Proof: According to the definition of the strict order, it is obvious that it is true. \square

FIGURE 7. Activity a and activity b are in exclusiveness which there is a path from a to b .

Proposition 3: For any sound free-choice system it holds that, if activity a and activity b are in exclusiveness relation based on log, then exclusiveness and structural exclusiveness coincide.

Proof: Assume activity a and activity b are not in structural exclusiveness. As $a +_L b$, it is impossible that activity a and activity b are in structural concurrent or structural cycle. If a path from activity a to activity b exists, from $a +_L b$, we know that the possible structure between them is shown as Figure 7. i.e. $\exists c$, satisfy $a \cap {}^\bullet c \neq \emptyset$, $c^\bullet \cap {}^\bullet b \neq \emptyset$. Clearly, activity b is not in choice-free structure, which leads to a contradiction. \square

Proposition 4: If activity a and activity b are in the concurrent interleaving order relation based on log, then they are in structural concurrent.

Proof: We assume activity a and activity b are not in concurrent structure in process model. From the definition of concurrent interleaving order relation, we know that activity a and activity b are either in exclusive structure or in sequential structure, so we have to consider two cases. For the former, without losing generality, we only consider the concurrent relation of length 1. In order to guarantee $a \geq_L b$, $b \geq_L a$ both hold and the system is sound, the possible structures are as shown in Figure 8. It is obviously that $b \geq_L a$, $a \geq_L b$ also hold except $a \geq_L b$, $b \geq_L a$. That yields a contradiction with $a ||_L b$. For the latter, as they are in sequential structure, then the possible structures are as shown in Figure 9, i.e. $t_1 \cap t_2^\bullet \neq \emptyset \wedge t_2^\bullet \cap a^\bullet \neq \emptyset \wedge t_1 \rightarrow_L t_2$. It is obviously that $b \geq_L a$ and $a \geq_L b$ both hold. That yields a contradiction with $a ||_L b$. As both assumptions lead to contradictions, we know they are structural concurrent. \square

Let T be set of activities and L be an event log over T . For $\sigma \in L$, the number of occurrences of activity a in the trace σ is denoted as $OccurTime(a, \sigma)$. For $T' \subset T$, the projection of a trace $\sigma \in L$ onto T' is denoted as $\sigma_{T'}$.

Proposition 5: If activity a and activity b are in the loop interleaving order relation based on log, we denote $T' = \{a, b\}$, then the structure between activity a and activity b is divided into three cases.

For convenience, only the most basic possible structure behavior relationship is considered here. Other complex behavior relationships can be transformed into the following basic structures by clustering equivalence classes.

- (1) If $\exists t_1, t_1 \rightarrow_L a, t_1 \rightarrow_L b$ $BD_{\min}(t_1, b, L)_{\min} \geq BD_{\min}(t_1, a, L)_{\min} + 1$ and $\forall \sigma \in L$,

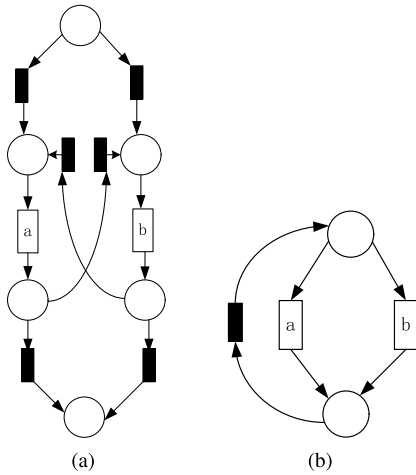


FIGURE 8. Activity a and activity b are in exclusive structure with interleaving order relation.

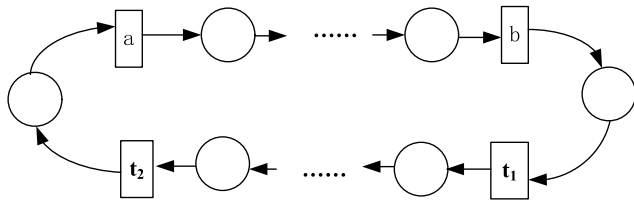


FIGURE 9. Activity a and activity b are in sequential structure with interleaving order relation.

$|OccurTime(a, \sigma_{T'}) - OccurTime(b, \sigma_{T'})| \leq 1$ both hold, then activity a and activity b are in sequential relation inside cyclic structure.

- (2) If $\exists t_1, t_1 \rightarrow_L a, t_1 \rightarrow_L b$ and $BDis(t_1, a, L)_{\min} = BDis(t_1, b, L)_{\min} = 1$ hold. Moreover, for $\forall \sigma \in L$, there is no dependencies between $OccurTime(a, \sigma_{T'})$ and $OccurTime(b, \sigma_{T'})$, i.e. the occurrence frequency between a and b is independence, then activity a and activity b are exclusive inside cyclic structure.
- (3) If $\exists t_1$, there is $t_1 \rightarrow_L a, t_1 \rightarrow_L b$, $BDis(t_1, a, L)_{\min} = BDis(t_1, b, L)_{\min} = 1$ and $BDis(t_1, a, L)_{\max} = BDis(t_1, b, L)_{\max} = 2$ Moreover, $\forall \sigma \in L$, for $OccurTime(a, \sigma_{T'}) = OccurTime(b, \sigma_{T'})$, then activity a and activity b are concurrent inside cyclic structure.

Proof: The above three cases are respectively proved. Without loss of generality, only the basic structural relations among activities are considered here, and other complex relations can be transformed into basic structural relations through clustering. \square

- (1) Case 1: As $\exists t_1, t_1 \rightarrow_L a, t_1 \rightarrow_L b$ hold, meanwhile, activity a and activity b are exclusive inside cyclic structure. The possible structures are shown in Figures 10(a), 10(b), 10(c). In Figure 10(a), $BDis(t_1, b, L)_{\min} \geq BDis(t_1, a, L)_{\min} + 1$ doesn't hold when the occurrence sequence in the form of $\dots t_1 b a \dots$, which leads to a contradiction. In Figure 10(b), $\forall \sigma \in L, |OccurTime(a, \sigma_{T'}) - OccurTime(b, \sigma_{T'})| \leq 1$

doesn't hold when the occurrence sequence in the form of $\dots t_1 a a a a \dots$, which leads to a contradiction. In Figure 10(c), $BDis(t_1, b, L)_{\min} \geq BDis(t_1, a, L)_{\min} + 1$ holds. When activity a and activity b are both in loop-body or loop-redo, $\forall \sigma \in L, OccurTime(a, \sigma_{T'}) = OccurTime(b, \sigma_{T'})$ holds. When activity a is in loop-body and activity b is in loop-redo, $\forall \sigma \in L, OccurTime(a, \sigma_{T'}) = OccurTime(b, \sigma_{T'}) + 1$ holds. Therefore $\forall \sigma \in L, |OccurTime(a, \sigma_{T'}) - OccurTime(b, \sigma_{T'})| \leq 1$ always holds in the Figure 10(c). Hence, for case 1, activity a and activity b must be sequential relation inside a cyclic structure.

- (2) Case 2: As $\exists t_1, t_1 \rightarrow_L a, t_1 \rightarrow_L b$, $BDis(t_1, a, L)_{\min} = BDis(t_1, b, L)_{\min} = 1$, The possible structures are shown in Figures 10(a), 10(b). In Figure 10(a), due to activity a and activity b are in concurrent structure, $\forall \sigma \in L, OccurTime(a, \sigma_{T'}) = OccurTime(b, \sigma_{T'})$ holds which leads to a contradiction. In Figure 10(b), as activity a and activity b are exclusive inside a cycle structure. Obviously, activity a and activity b can occur at any number of times, which make the assume hold true. Hence, for case 2, activity a and activity b must be exclusive relationship in a cyclic structure.
- (3) For case 3, as $\exists t_1, t_1 \rightarrow_L a, t_1 \rightarrow_L b$, $BDis(t_1, a, L)_{\max} = BDis(t_1, b, L)_{\max} \geq 2$, $BDis(t_1, a, L)_{\min} = BDis(t_1, b, L)_{\min} = 1$ hold, the possible structures are shown in Figures 10(a), 10(b), 10(d). In Figure 10(a) and 10(d), there exists infinite occurrence sequence in form of $\dots t_1 a b \dots, \dots t_1 b a \dots, \dots t_1 a b b a \dots, \dots t_1 b a a b \dots, \dots$, which satisfy $|BDis(t_1, a, L)_{\min} - BDis(t_1, b, L)_{\min}| = 1$ and for $\forall \sigma \in L, OccurTime(a, \sigma_{T'}) = OccurTime(b, \sigma_{T'})$. Therefore, the assume holds true. In Figure 10(b), as activity a and activity b are in exclusive relation inside the loop structure, the numbers of occur times between them is independent which yields a contradiction. Hence, for case 3, activity a and activity b must be concurrent relationship inside cyclic structure.

Lemma 1: If activities are in concurrent interleaving order relation, then there are silent transitions of and-split and and-join type.

Proof: Obviously it is correct. \square

Although the silent transition of and-gateway type added by Lemma 1 will not affect the behavior relation of any others activities of the net system, it may produce many redundant silent transitions of and-gateway type. Algorithm 1 will remove these redundant silent transitions of the and-gateway type according to structural features between the silent transition and its pre - or post-transitions.

Theorem 1: Let L be the event log over T . For $T' \subset T$, if T' is a maximal subset of activities that is in the loop interleaving order relation and $\forall t \in T'$ belongs to the part of loop-body, then there exists a silent transition of loop type that belongs to the part of loop-redo.

Proof: We assumed that there is no silent transition of loop type, i.e., there exists a visible activity t' as a loop-redo

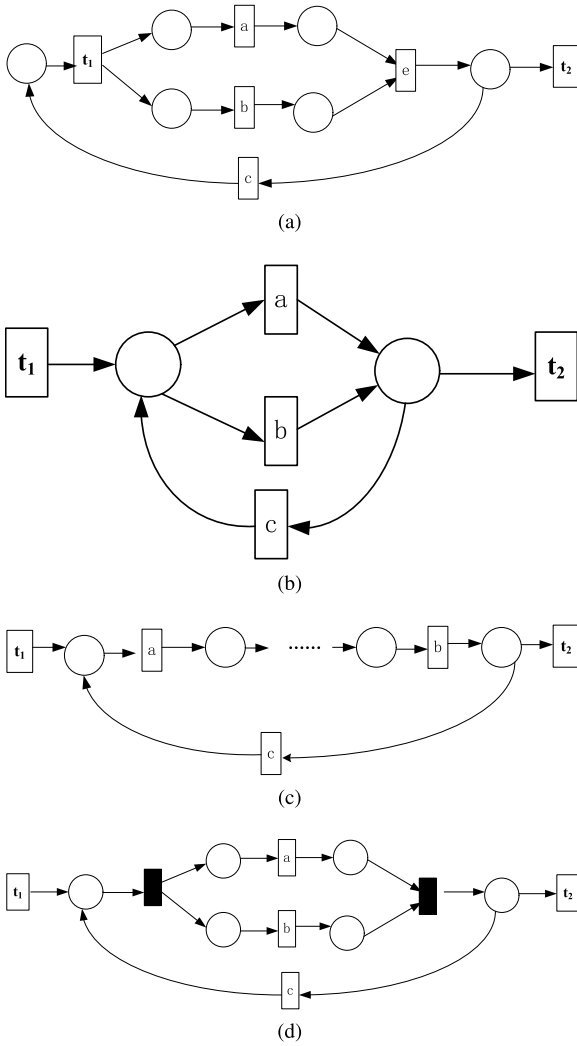


FIGURE 10. The possible structures between activity *a* and activity *b* when they are in the loop interleaving order relation.

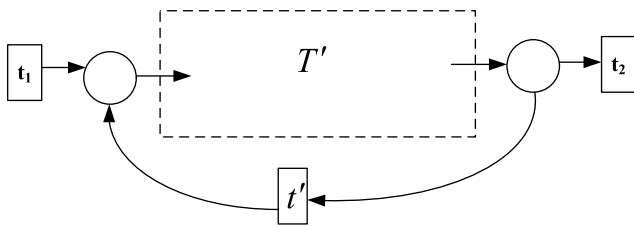


FIGURE 11. A visible activity as a loop-redo transition.

transition. As $\forall t \in T'$ belongs to the part of the loop-body, the possible net construction of T' and t' is shown in Figure 11. It is easy to know that $\forall t \in T'$ and t' are in the loop interleaving order relation. Hence, the maximal subset of activity that is in the loop interleaving order relation is $T' \cup t'$, which contradict with the condition. Therefore, Theorem 1 is proven to be correct. \square

Silent transitions of side type are no longer discussed here, as for models with multiple start or end activities, they can be

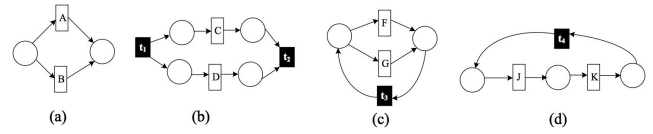


FIGURE 12. Sub-modules.

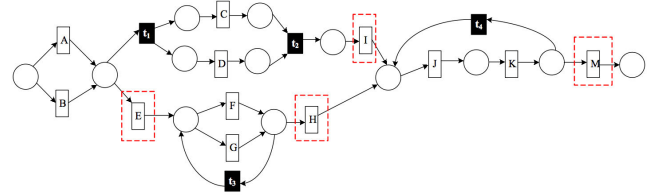


FIGURE 13. An initial Petri net with silent transitions of and-gateway and loop type.

realized only by manually adding a unique start or a unique end transition. Algorithm 1 presents how to construct an initial process model with silent transitions of and-gateway type and loop type from the event log. First of all, the behavior distance matrix and behavior relationship matrix between activities is obtained according to the event log L , after that several sub-modules are constructed based on the basic behavior relationship. Subsequently, these sub-modules are merged into a whole process model.

Meanwhile, silent transitions of and-gateway and loop type can be discovered using Lemma 1 and Theorem 1. Finally, the redundant silent transitions of and-gateway types are removed from the process model.

SubSet and *subModules* are activity subset getting from a behavior relation Matrix and sub-module set composed of these activity subsets respectively. Function *Construct*(S , T , *structural exclusiveness*) construct a sub-module with S and T in structural exclusiveness, and others are similar. *InsertSilent*(*and* – *type*) discovers a silent transition of and-type, and others are similar. *Merge*(Md_i , Md_j , Md_{ij}) combine sub-modules Md_i and Md_j to get a larger sub-module Md_{ij} . Step1-Step2 creates a behavior distance Matrix and a behavior relation Matrix based on event logs. Step4-Step30 constructs several sub-modules with silent transitions consisting of activity subsets. In Step31-Step36, the sub-modules are combined into a complete process model through causal dependency and strict relation in the behavior relation matrix. Step37-45 removes the redundant silent transitions of and-gateway type.

The execution of Algorithm 1 is illustrated using the event log L_1 in Section II. The behavior distance matrix and basic behavior relationship matrix are shown in Figure 4 and Figure 5, respectively, in Section II. The four sub-modules shown in Figure 12 are constructed according to Step4-30. Subsequently, they are merged into a whole process model by using Step31-36, as shown in Figure 13.

Finally, according to Step37-45, the redundant silent transitions of and-gateway types in Figure 11 are removed, and the process model with silent transitions of and-gateway type

Algorithm 1 The Method for Constructing a Free-Choice Petri Net With Silent Transitions of And-Gateway Type and Loop Type

Input: The event log L
Output: The free-choice Petri net M_0

```

1 Create  $BDisMatrix_{n \times n}$ ;
2 Create  $BRelMatrix_{n \times n}$ ;
3  $subModules = \phi$ ,  $SubSet = \phi$ ;
4  $SubSet = \text{GetsubActivitySet}(BRelMatrix_{n \times n}) = \{subM_1, subM_2, \dots, subM_k\}$ 
5 foreach  $subM_i$  in  $SubSet$  do
6   if  $S, T \in subM_i \wedge S +_L T$  then
7      $Md = \text{Construct}(S, T, \text{structural exclusiveness})$ 
8   end
9   if  $S, T \in subM_i \wedge S \parallel_L T$  then
10     $Md = \text{Construct}(S, T, \text{structural concurrent}) \wedge \text{InsertSilent}(\text{and - type})$ 
11  end
12  if  $S, T \in subM_i \wedge S \parallel'_L T$  then
13     $\text{replace}(S, s', \sigma), \text{replace}(T, t', \sigma), \exists a \in \mathcal{A} \text{ such that } a \rightarrow_L s', a \rightarrow_L t' \text{ holds}$ 
14    for  $\forall a \in T, T' \in T$  do
15      if  $BDis(a, s', L)_{\min} \geq BDis(a, t', L)_{\min} + 1$  and for  $T' = S \cup T, \forall \sigma \in L, BDis_{\min}(s', t', L) \geq 2$  then
16         $Md = \text{Construct}(S, T, \text{sequential relation inside a loop structure})$ 
17      end
18      if  $BDis(a, s', L)_{\min} = BDis(a, t', L)_{\min} = 1$  and for  $\forall \sigma \in L, \text{OccurTime}(s', \sigma_{T'})$  and  $\text{OccurTime}(t', \sigma_{T'})$ 
        have no dependencies then
19         $Md = \text{Construct}(S, T, \text{exclusive relation inside a loop structure})$ 
20      end
21      if  $BDis(a, s', L)_{\min} = BDis(a, t', L)_{\min} = 1, BDis(a, s', L)_{\max} = BDis(a, t', L)_{\max} = 2$  and for
         $\forall \sigma \in L, \text{OccurTime}(s', \sigma_{T'}) = \text{OccurTime}(t', \sigma_{T'})$  then
22         $Md = \text{Construct}(S, T, \text{concurrent relation inside a loop structure})$ 
23      end
24    end
25    if  $S, T \in \text{Loop - body}$  then
26       $\text{InsertSilent}(\text{loop - type})$ 
27    end
28  end
29   $subModules = subModules \cup Md$ 
30 end

31 while  $|subModules| \neq 1$  do
32   foreach  $\rightarrow_L$  and  $\rightarrow_L$  in  $BRelMatrix$  do
33      $\text{Merge}(Md_i, Md_j, Md_{ij})$ 
34      $subModules = subModules - Md_i - Md_j + Md_{ij}$ 
35   end
36 end
37 for all silent transitions of and - split and and - join type do
38   let  $l(t) = \tau$ 
39   if  $t^\bullet > 1$  and  $|\bullet(t)| = 1, l(\bullet(t)) \neq \tau$  then
40      $\text{Replace}(t, \bullet(t))$ 
41   end
42   if  $\bullet t > 1$  and  $|(t)^\bullet| = 1, l((t)^\bullet) \neq \tau$  then
43      $\text{Replace}(t, (t)^\bullet)$ 
44   end
45 end
46 return A free - choice Petri net with silent transition of and - gateway and loop type
  
```

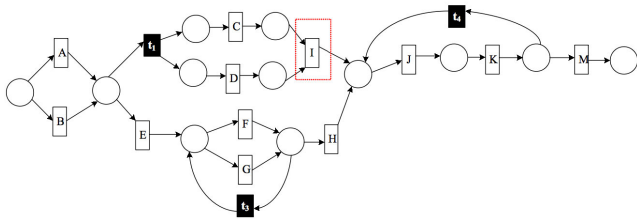


FIGURE 14. The process model with silent transitions of and-gateway type and loop type without redundant silent transitions M_0 .

and loop type without redundant silent transitions is built, as shown in Figure 14.

VI. MINING SILENT TRANSITIONS OF SKIP TYPE

Section IV has shown how to construct a process model with silent transitions of and-gateway type and loop type. This section discusses how to discover silent transitions of skip type based on the initial process model constructed by Algorithm 1, thereby further optimize the initial process model.

Definition 10 (Model-Based Minimum Behavior Distance): Let $S = (N, M_0)$ be a net system, where $N := (P, T, F, \lambda)$ is a labeled Petri net, $\mathcal{A} \cup \tau$ is all transition labels. For $\forall a, b \in \mathcal{A}$, model-based minimum behavior distance between a and b is defined as

$$BD_{\min}(a, b, S) = \{k | \forall \sigma \in \mathcal{T}(S), BD_{\min}(a, b, \sigma) \geq k\}. \quad (4)$$

Definition 10 denotes that if activity a and b occur in executable traces of the model, there are at least k -activities between them. For instances, in Figure 14 $BD_{\min}(E, H, S) = 2$, $BD_{\min}(A, I, S) = 3$.

Theorem 2: Let L be an event log over \mathcal{A} and $S = (N, M_0)$ be a net system, where $N := (P, T, F, \lambda)$ is a labeled Petri net. $\forall a, b \in \mathcal{A}$, if $BD_{\min}(a, b, L) = 1$, $BD_{\min}(a, b, S) > 1$ and $a \rightarrow_L b$, then there exists a silent transition of skip type between a and b .

Proof: We assumed that there is no silent transition of skip type between a and b . As $a \rightarrow_L b$, there exists a directed path consisting of places and transitions from a to b . In addition, $BD_{\min}(a, b, S) > 1$ also holds. Hence, there is at least one visible transition between a and b , so $\nexists \sigma \in \mathcal{T}(S)$ such that $BD_{\min}(a, b, \sigma) = 1$, i.e. $BD_{\min}(a, b, L) = 1$ is not correct, which obviously contradicts the condition. As a consequence, Theorem 2 holds. \square

Definition 11 (Pre- and Post-Transition Sets): Let $S = (N, M_0)$ be a net system, where $N := (P, T, F, \lambda)$ is a labeled Petri net. $\forall t \in T$, $\bullet(t)_T = \{t' | t' \in T \wedge (t')^\bullet \cap \bullet t \neq \emptyset\}$, $\bullet(t)_T$ is called a pre-transition set of t . $(t)^\bullet_T$ is called a post-transition set of t , which is defined similarly.

Definition 12 (Model-Based Minimum Behavior Distance of Concurrent Structure): Let $S = (N, M_0)$ be a net system, where $N := (P, T, F, \lambda)$ is a labeled Petri net. $t_1, t_2 \in T$, t_1 is an and-split transition, t_2 is an and-join transition.

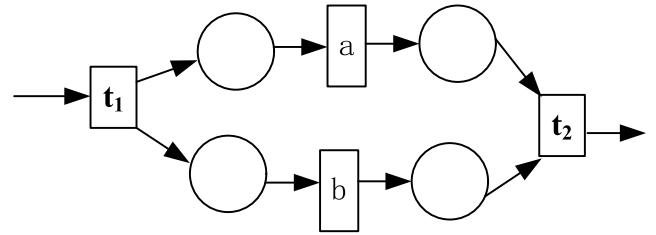


FIGURE 15. A substructure containing visible and-gateway transitions.

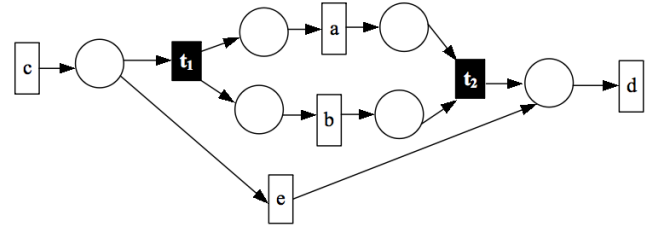


FIGURE 16. A substructure containing silent transitions.

The set of all executable transition sequences of S is denoted as $\mathcal{T}(S)$, Model-based minimum behavior distance of concurrent structure between t_1 and t_2 is defined as:

$$BD_{\min}^{\parallel}(t_1, t_2, S) = \{k | \forall \sigma \in \mathcal{T}(S) \wedge BD_{\min}(t_1, t_2, \sigma) \geq k\} \quad (5)$$

According to Definition 12, we can get $BD_{\min}^{\parallel}(f, i, S) = 1$ in Figure 18(b) and $BD_{\min}^{\parallel}(f, j, S) = 2$ in Figure 18(d).

Theorem 2 cannot solve the silent transition in concurrent structures. Theorem 3 presents a method for mining silent transitions in concurrent structures to solve this limitation.

Theorem 3: Let L be an event log over \mathcal{A} and $S = (N, M_0)$ be a net system, where $N := (P, T, F, \lambda)$ is a labeled Petri net. $\forall t_1, t_2 \in T$, where t_1 is an and-split transition, t_2 is an and-join transition. There exists a silent transition of skip type on concurrent branches between t_1 and t_2 , if one of the following conditions holds:

$$(a) \quad BD_{\min}(\lambda(t_1), \lambda(t_2), L) < BD_{\min}^{\parallel}(t_1, t_2, S), \quad \text{where } \lambda(t_1) \neq \tau, \lambda(t_2) \neq \tau \quad (6)$$

$$(b) \quad \exists \sigma \in L, a \in \sigma \vee b \in \sigma \wedge BD_{\min}(c, d, \sigma) < BD_{\min}^{\parallel}(t_1, t_2, S), \quad \text{where } \lambda(t_1) = \lambda(t_2) = \tau, \quad \forall c \in \bullet(t_1)_T, \quad \forall d \in (t_2)_T^\bullet, a, b \in (t_1)_T^\bullet \quad (7)$$

Proof: Without loss of generality, only the concurrency structure of the most basic case is given to prove the correctness of the theorem. Let us assume that there exists no silent transition of skip type. We discuss it in two cases:

(1) when t_1 and t_2 are both visible transitions, the corresponding possible substructure is shown in Figure 15. $\forall \sigma \in \mathcal{T}(S)$, if $t_1 \in \sigma$, then σ is the form of $\dots t_1 a b t_2 \dots$ or $\dots t_1 b a t_2 \dots$. Clearly, $BD_{\min}(t_1, t_2, S) = 3$. If there is no silent transition on the concurrent structure between t_1 and t_2 , then $BD_{\min}(\lambda(t_1), \lambda(t_2), L) \geq 3$ that contradicts the conditions of $BD_{\min}(\lambda(t_1), \lambda(t_2), L) < BD_{\min}^{\parallel}(t_1, t_2, S)$.

Algorithm 2 A Method to Discover Silent Transitions of Skip Type

Input: An event log L , an initial Petri net M_0 obtained by Algorithm 1, the behavior distance matrix $BDisMatrix_{n \times n}$

Output: An Petri net with multiple types of silent transitions M_1

```

1 traverse  $BDisMatrix_{n \times n}$ 
2 foreach  $a, b \in \mathcal{A}$  do
3   if  $BDis_{\min}(a, b, L) = 1, BDis_{\max}(a, b, L) \neq 1$  then
4      $Getvalue(BDis_{\min}(a, b, S))$ 
5     if  $BDis_{\min}(a, b, S) > 1$  then
6        $InsertSilent(a, b, skip - type)$ 
7     end
8   end
9 end
10 foreach transitions of and-split and and-join do
11   let  $t_1$  be an and - split transition,  $t_2$  be an and - join transition
12   if  $\lambda(t_1) \neq \tau \wedge \lambda(t_2) \neq \tau \wedge BDis_{\min}(t_1, t_2, L) \neq BDis_{\max}(t_1, t_2, L)$  then
13      $Getvalue(BDis_{\min}^{\parallel}(t_1, t_2, S))$ 
14     if  $BDis_{\min}(t_1, t_2, L) < BDis_{\min}^{\parallel}(t_1, t_2, S)$  then
15        $InsertSilent(t_1, t_2, skip - type)$ 
16     end
17   else
18     if  $\lambda(t_1) = \tau \wedge \lambda(t_2) = \tau$  then
19       let  $\forall c \in \bullet(t_1)_T, \forall d \in (t_2)_T^{\bullet}, a, b \in (t_1)_T^{\bullet}$ 
20       if  $BDis_{\min}(c, d, L) \neq BDis_{\max}(c, d, L)$  then
21          $Getvalue(BDis_{\min}^{\parallel}(t_1, t_2, S))$ 
22         if
23            $\exists \sigma \in L, a \in \sigma \vee b \in \sigma \wedge BDis_{\min}(c, d, \sigma) < BDis_{\min}^{\parallel}(t_1, t_2, S)$  then
24            $InsertSilent(t_1, t_2, skip - type)$ 
25         end
26       end
27     end
28 end
29 return A Petri net with multiple types of silent transitions

```

(2) when t_1 and t_2 are both silent transitions, the corresponding possible substructure is shown in Figure 16, $\forall \sigma \in T(s)$, if $a \in \sigma \vee b \in \sigma$, then an executable trace is the form of $\sigma = \dots ct_1abt_2d \dots$ or $\sigma = \dots ct_1bat_2d \dots$, hence $BDis_{\min}^{\parallel}(t_1, t_2, S) = 3, \forall \sigma \in L$, if, then $BDis(c, d, \sigma) \geq 3$. Therefore, if there is no silent transition on the concurrent structure between t_1 and t_2 , it is impossible to make $BDis_{\min}(c, d, \sigma) < BDis_{\min}^{\parallel}(t_1, t_2, S)$ hold. As a consequence, Theorem 3 holds. \square

Based on Algorithm1, Algorithm 2 presents a method to discover silent transitions of skip type. The basic idea of Algorithm 2 is to compare the minimum behavior distance

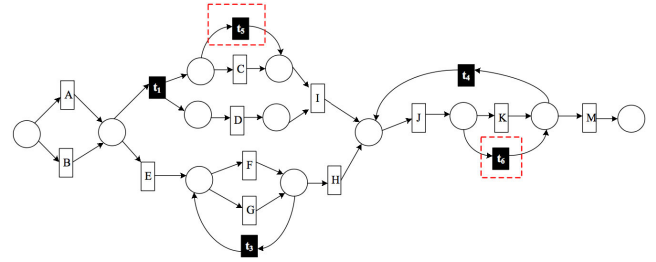


FIGURE 17. A Petri net M_1 with multiple types of silent transition.

of activities in the log with their minimum behavior distance in the model, and determine whether there is a silent transition of skip type between them using Theorem 2 and Theorem 3.

For the initial Petri net model M_0 in Figure 13, according to Step2-9 of Algorithm 2, we can find a silent transition of skip type between activity pairs (J, M) . The silent transition t_5 between A and I can be discovered using step 18-23 of Algorithm 2. Finally, an optimized Petri net M_1 with multiple types of silent transitions is obtained, as shown in Figure 17.

VII. EXPERIMENTS

In this section, series of experiments with real and synthetic event logs have been conducted to assess the goodness of our approach. Section A presents the evaluation criteria used in this paper. Section B presents the synthetic event logs used in this section. In Section C, we compare the proposed method with existing approaches in terms of the number of silent transitions discovered and the quality of the process model using various synthetic event logs. In Section D, we discuss the experimental result on real event logs.

A. EVALUATION CRITERIA

Determining the quality of a process mining result is characterized by many dimensions. Usually, four main quality dimensions are used: fitness, precision, simplicity, generalization. In this paper, we use two main and most important quality criteria: fitness, precision. Fitness is to measure how much behavior in the event log can be replayed by the model. A model with good fitness allows for the behavior seen in the event log. The model has a perfect fitness if all traces in the log can be replayed by the model from beginning to end. There are various ways of defining fitness. Here, we use align-based technique to compute the fitness of the discovered model [24].

Precision is to measure how much behavior the model produces that is not observed in the event log. Here, we use the method proposed in [25] to calculate the precision. The model having a low precision will lead to the underfitting phenomenon. Therefore, a good mining algorithm should score well on the precision of the process model as can as possible when the fitness is good. Sometimes, putting aside a small amount of behavior causes a slight decrease in the fitness value, whereas the precision value increases much more. Therefore, we use the F-measure that combines fitness

and precision as follows [26].

$$F - measure = \frac{2 \times Precision \times Fitness}{Precision + Fitness} \quad (8)$$

Rediscovery mainly concerns whether the discovered process model is equivalent to the original model. When comparing two models, we only consider the behavioral relationship between them in this paper. However, the traditional trace equivalence yields a true or false answer and can, therefore, not be directly applied if models overlap partially. It cannot intuitively reflect what extent behavioral equivalence is between two models. Therefore, this paper quantifies the degree of consistency between the mined model and the original model using the trace consistency measurement proposed in literature [23].

B. SYNTHETIC EVENT LOGS

The event logs used in this section are mainly from two aspects. One is the previous synthetic event log from Section II, and the other is generated by simulating a known process model, which contains multiple silent transitions of different types. For real event logs, there is no reference process model available to compare with the results of process discovery algorithms. Moreover, the correct number of silent transitions in the process model is also uncertain. Therefore, we designed fifteen artificial process models with different behavior and different types of silent transitions. The maximum number of activities in one process model is less than 20. 15 groups of event logs are generated by simulating fifteen artificial process models. Therefore, unlike real event logs, for each synthetic event log there is a corresponding reference process model. We can use these reference models as the ground truths that indicate how many silent transitions exist in the original models. Parts of the models are shown in Figure 18. The first model contains silent transitions of skip type, the second one contains silent transitions of loop and skip type, the third model contains silent transitions of side, and-gateway and skip type, the fourth one contains many silent transitions of skip type and loop, the fifth process model contains silent transitions of loop, and-gateway and skip type, and the last one contains all the previous silent transitions.

C. EVALUATION RESULTS WITH SYNTHETIC EVENT LOGS

For the previous synthetic event log and the 15 groups of event logs generated artificially, the number of silent transitions, the fitness of the model [21], the precision [22], and the consistency between the discovered model and the original model [23] are compared respectively.

The comparison results based on the event log L_1 are depicted in Table 1. The experimental results indicate that the proposed approach is more superior than others in terms of the number of silent transitions, the fitness and the precision of the model.

We apply IM algorithm, $\alpha \#$, and the proposed method on the 15 groups of event logs generated artificially and give a comparison result to evaluate the proposed method.

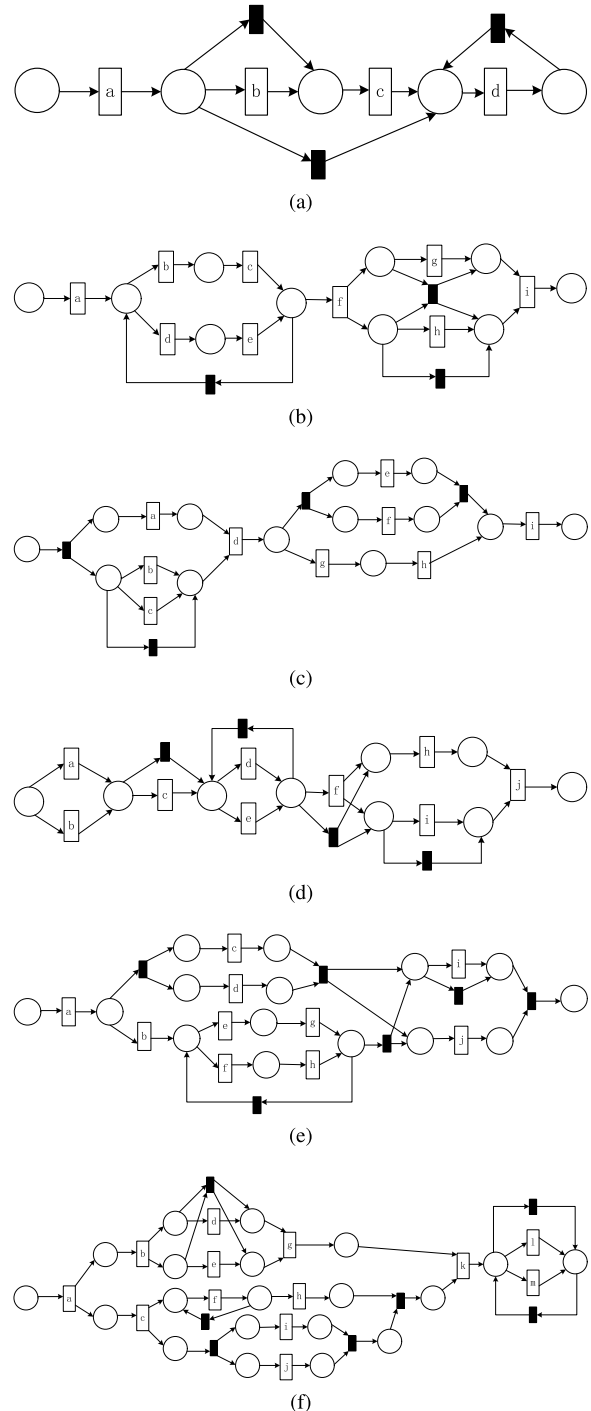


FIGURE 18. Synthetic process models with different types of behavior and silent transition.

For the event log generated by model 1 shown in Figure 18(a), when applying the proposed method and $\alpha \#$ algorithm on this event log, three silent transitions in the original model can be rediscovered. Whereas IM algorithm produces four silent transitions, it discovers two silent transitions on the path from activity B to activity D instead of one silent transition. Therefore, the process model constructed by IM algorithm will have more additional behaviors. For the event

TABLE 1. The comparison results based on the event log.

Name	The proposed approach	α # algorithm	IM algorithm
The number of silent transitions	5	3	12
fitness	1	0.975	1
precision	1	1	0.962

log generated by model 2 shown in Figure 18(b), three silent transitions can be found using the proposed method and α # algorithm. However, the difference is that α # algorithm inserts a silent transition between the pre-set place and post-set place of G, thus the discovered model generates an additional executable sequence in the form of ‘... fhi...’. Five silent transitions with the redundant silent transition of and-type are discovered by IM algorithm. For the event log generated by model 3 shown in Figure 18(c), the proposed method and IM algorithm both can rediscover four silent transitions included in the original model. Meanwhile, the model discovered by them has a perfect fitness and precision. However, the α # algorithm only discovers one silent transition of side-type and the fitness of the discovered process is less than 1. For the event log generated by model 4 shown in Figure 18(d), the proposed method discovers 4 silent transitions included in the original model. The α # algorithm discovers 3 silent transitions and fails to discover the silent transition occurred in the concurrent structure, which leads to the fitness less than 1. However, the model discovered by IM algorithm produces 14 silent transitions, which leads to the precision less than 1. However, for the event logs generated by Figures 18(a), 18(b), 18(c) and 18(d), the proposed method can all rediscover the original model and have perfect fitness and precision. For the event log generated by model 5 shown in Figure 18(e), the proposed method can discovered 6 silent transitions, however the α # algorithm can only discover 1 silent transition, which leads to the fitness and precision less than 1. However, IM algorithm and the proposed method discover 7 silent transitions and has a perfect fitness and precision. For the event log generated by model 6 shown in Figure 18(f), the proposed method discovers 8 silent transitions, α # algorithm can only discover 4 silent transitions which leads to the precision and fitness are both less than 1. However the model obtained by IM algorithm includes 5 silent transitions, which leads to the fitness less than 1.

Figures 19(a), 19(b), 19(c), 19(d) and 19(e) show the comparison results of IM algorithm, α # algorithm, and the proposed method in terms of the number of silent transitions, the fitness, the precision, the consistency and F-measure. Figure 19(a) indicates that the method proposed in this paper can identify multiple types of silent transitions in the original model. The number of silent transitions lies between the number of the IM algorithm and the α # algorithm without too many redundant silent transitions.

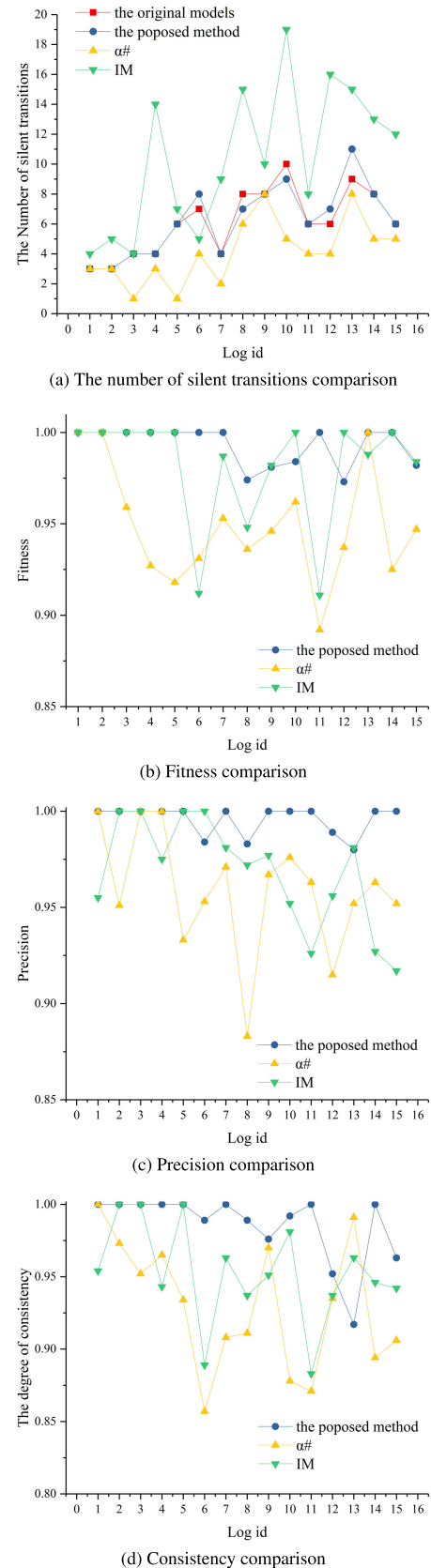
**FIGURE 19.** The results of the comparison on 15 logs.

TABLE 2. Basic information of these event logs.

Event log	#Case	#Event	#activity	#min-event-per-trace	#max-event-per-trace
BPIC_2012	13087	262200	36	3	175
BPIC_2013_incidents	7554	65533	13	1	123
BPIC_2013_open	819	2351	5	1	22
BPIC_2014_close	1487	6660	7	1	35

TABLE 3. The experiment results applying different real-life event logs and discovery technique.

Process Discovery	Metric	BPIC_2012	BPIC_2013_incidents	BPIC_2013_open	BPIC_2013_close
alpha#	Fitness	1	0.845	1	0.897
	Precision	0.127	0.657	0.325	0.564
	F-measure	0.225	0.739	0.491	0.693
	#Silent Transition	6	4	3	5
IM	Fitness	0.867	0.816	1	1
	Precision	0.798	0.869	0.679	0.344
	F-measure	0.831	0.842	0.809	0.512
	#Silent Transition	36	50	22	22
the proposed method	Fitness	0.923	0.928	0.916	0.864
	Precision	0.867	0.875	0.873	0.823
	F-measure	0.894	0.901	0.894	0.843
	#Silent Transition	16	26	18	12

Figures 19(b) and 19(c) indicate that the proposed method may cause a little decrease in fitness value in some situation, yet always yield a notable increase in precision value compare with others. Figure 19(d) shows that compared with the IM algorithm and α # algorithm, the behavior of the model obtained by this method is more consistent with that of the original model. Figure 19(e) shows that the F-measure of the obtained process models by the proposed approach has a better result than the others. In conclusion, the experimental results show that the proposed method can find multiple types of silent transitions. Meanwhile, it improves the quality of process discovery results.

D. EVALUATION USING REAL EVENT DATA

In this section, we discuss the result using the real logs from the Business Process Intelligence Challenge(BPIC), which are accessible via *4tu Center for Research Dat* ¹. Table 2 reports the characteristics of all logs used in terms of number of case, number of events, number of unique labels minimum number of events per trace, maximum number of events per trace for each log. We mined the process models with three different discovery algorithms, the proposed approach, IM algorithm, α # algorithm used 4 BPIC logs. Results of this experiment are given in Tab. 3.

¹https://data.4tu.nl/articles/dataset/BPI_Challenge_2012/12689204;
https://data.4tu.nl/articles/dataset/BPI_Challenge_2013_incidents/12693914 https://data.4tu.nl/articles/dataset/BPI_Challenge_2013_open_problems/12688556 https://data.4tu.nl/articles/dataset/BPI_Challenge_2013_closed_problems/12714476

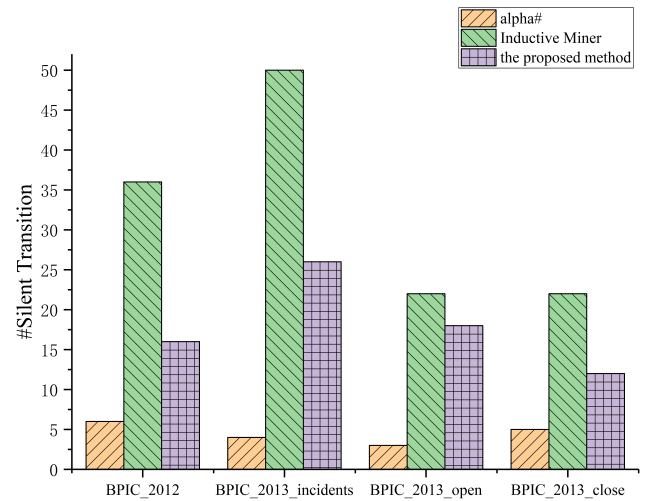
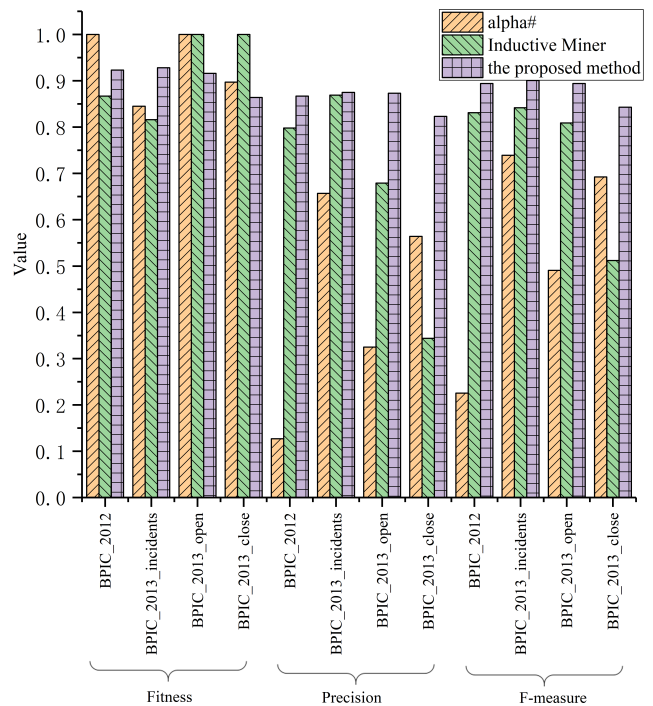
**FIGURE 20.** The number of silent transition comparison using different real-life logs and discovery algorithm.**FIGURE 21.** Fitness, Precision, F-Measure comparison using different real-life logs and discovery algorithm.

Figure 20 shows the result of silent transition, obtained when applying three techniques on different real-life event logs. As we see, for these four real event logs, the proposed method produces fewer redundant silent transitions than the IM algorithm. Moreover, the number of silent transitions mined by the proposed method lies between the other two algorithms.

Figure 21 shows the obtained fitness, precision, F-measure values of applying different methods on real event logs. For BPIC_2012 and BPIC_2013_Open event log, the process

model obtained by α # algorithm has perfect fitness. However, the precision of the model is poor as it contains several isolated activities which can produce arbitrary behaviors. For BPIC_2013_Open and BPIC_2013_Close event log, the process model constructed by IM algorithm both has perfect fitness, but due to the existence of redundant transitions, loops, and concurrency, its precision is very poor, especially BPIC_2013_Close event log, which can produce executable sequences of any length. For BPIC_2013_Incidents event log, the model generated by the IM algorithm contains 50 silent transitions, including a large number of redundant silent transitions. Compared with the IM algorithm, the process model obtained by the proposed method achieves more high fitness and precision, as well as reduces the number of redundant silent transitions. For BPIC-2012 event log, the process model generated by α # algorithm contains a large number of isolated activities, that resulting in very low precision of the model. Although the process models built by other algorithms have perfect fitness, however the precision is very poor because of these models produce too many extra behaviors not in event log. Hence, the quality of the obtained process model is still relatively poor. In general, it is more reasonable to use F-measure to measure the quality of the process model. As shown in Figure 21, for all event logs, the proposed method can discover a process model with a high F-measure value.

Based on the experiments using real event log, we conclude that the proposed method presents a little drop in fitness and a significant increase in precision. However, the F-measure significantly improves when our technique is used compared the others. This significant increment of F-measure is explained by the noticeable and significant increment of precision. Therefore, the result from synthetic event logs and real-life event logs both confirm the effectiveness of our technique.

VIII. CONCLUSION AND FUTURE WORK

The paper proposes a novel method to mine process models with silent transitions based on behavior distance. First, to more accurately capture the dependencies between activities, the minimum and maximum behavior distances of activities based on the event log are calculated. And then, the basic behavior relationship between pairs of activities based on the event log is obtained through the weak order relationship and behavior distance between activities. After that, an initial process model with silent transitions of and-gateway type and loop type is constructed by analyzing the basic behavior relationship and behavior distance. Based thereon, silent transitions of skip type are mined using behavior distance between activities based on log, model, and concurrent structure. Finally, lots of experiments have been done to verify the effectiveness of the proposed method. The experimental results show that the proposed method can successfully discover various types of silent transitions and improve the model's F-measure without significantly reducing the fitness of the model. The future work mainly has two aspects: (1) considering how to improve the method to extend it to a

non-free choice structure. (2) When the behavioral relationship of the log is incompleteness, how to correctly discover the possible behavioral relationship between activities.

REFERENCES

- [1] W. van der Aalst, *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Heidelberg, Germany: Springer, 2011.
- [2] S. J. J. Leemans, D. Fahland, and W. M. P. van der Aalst, "Discovering block-structured process models from event logs—A constructive approach," in *Proc. Int. Conf. Appl. Theory Petri Nets Concurrency*. Berlin, Germany: Springer, 2013, pp. 311–329.
- [3] A. Weijters, W. M. P. van der Aalst, and A. K. A. De Medeiros, "Process mining with the heuristics miner-algorithm," Technische Univ. Eindhoven, Eindhoven, The Netherlands, Tech. Rep. 166, 2006, pp. 1–34.
- [4] 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," *Knowl. Inf. Syst.*, vol. 59, no. 2, pp. 251–284, 2019.
- [5] W. M. P. van der Aalst and A. J. M. M. Weijters, "Process mining: A research agenda," *Comput. Ind.*, vol. 53, no. 3, pp. 231–244, Apr. 2004.
- [6] C. Liu, H. Duan, Q. Zeng, M. Zhou, F. Lu, and J. Cheng, "Towards comprehensive support for privacy preservation cross-organization business process mining," *IEEE Trans. Services Comput.*, vol. 12, no. 4, pp. 639–653, Jul. 2019.
- [7] C. Liu, Q. Zeng, L. Cheng, H. Duan, M. Zhou, and J. Cheng, "Privacy-preserving behavioral correctness verification of cross-organizational workflow with task synchronization patterns," *IEEE Trans. Autom. Sci. Eng.*, vol. 18, no. 3, pp. 1037–1048, Jul. 2021.
- [8] C. Liu, Y. Pei, L. Cheng, Q. Zeng, and H. Duan, "Sampling business process event logs using graph-based ranking model," *Concurrency Comput., Pract. Exp.*, vol. 33, no. 5, pp. 1–14, Mar. 2021.
- [9] C. Liu, Q. Zeng, L. Cheng, H. Duan, and J. Cheng, "Measuring similarity for data-aware business processes," *IEEE Trans. Autom. Sci. Eng.*, early access, Jan. 26, 2021, doi: [10.1109/TASE.2021.3049772](https://doi.org/10.1109/TASE.2021.3049772).
- [10] L. Wen, J. Wang, and J. Sun, "Mining invisible tasks from event logs," in *Advances in Data and Web Management*. Berlin, Germany: Springer-Verlag, 2007, pp. 358–365.
- [11] L. Wen, J. Wang, W. M. P. van der Aalst, B. Huang, and J. Sun, "Mining process models with prime invisible tasks," *Data Knowl. Eng.*, vol. 69, no. 10, pp. 999–1021, Oct. 2010.
- [12] W. M. P. van der Aalst, A. K. A. de Medeiros, and A. J. M. M. Weijters, "Genetic process mining," in *Proc. Int. Conf. Appl. Theory Petri Nets*. Berlin, Germany: Springer, 2005, pp. 48–69.
- [13] X.-Q. Huang, L.-F. Wang, W. Zhao, S.-K. Zhang, and C.-Y. Yuan, "A workflow process mining algorithm based on synchro-net," *J. Comput. Sci. Technol.*, vol. 21, no. 1, pp. 66–71, Jan. 2006.
- [14] A. Rozinat and W. M. P. van der Aalst, "Decision mining in ProM," in *Proc. Int. Conf. Bus. Process Manage.* Berlin, Germany: Springer, 2006, pp. 420–425.
- [15] Q. Guo, L. Wen, J. Wang, Z. Yan, and P. S. Yu, "Mining invisible tasks in non-free-choice constructs," in *Proc. 13th Int. Conf. Bus. Process Manage. (BPM)*. Berlin, Germany: Springer, 2015, pp. 109–125.
- [16] R. Sarno and K. R. Sungkono, "Coupled hidden Markov model for process mining of invisible prime tasks," *Int. Rev. Comput. Softw.*, vol. 11, no. 6, pp. 539–547, 2016.
- [17] R. Sarno and K. R. Sungkono, "Coupled hidden Markov model for process discovery of non-free choice and invisible prime tasks," *Proc. Comput. Sci.*, vol. 124, pp. 134–141, Jan. 2017.
- [18] K. R. Sungkono and R. Sarno, "Constructing control-flow patterns containing invisible task and non-free choice based on declarative model," *Int. J. Innov. Comput., Inf. Control*, vol. 14, no. 4, pp. 1285–1299, 2018.
- [19] R. Sarno, K. Sungkono, R. Johanes, and D. Sunaryono, "Graph-based algorithms for discovering a process model containing invisible tasks," *Int. J. Intell. Eng. Syst.*, vol. 12, no. 2, pp. 85–94, Apr. 2019.
- [20] T. Murata, "Petri Nets: Properties, analysis and applications," *Proc. IEEE*, vol. 77, no. 4, pp. 541–580, Apr. 1989.
- [21] A. Polyvyanyy, W. M. P. V. D. Aalst, A. H. M. T. Hofstede, and M. T. Wynn, "Impact-driven process model repair," *ACM Trans. Softw. Eng. Methodol.*, vol. 25, no. 4, pp. 1–60, May 2017.
- [22] M. Weidlich, A. Polyvyanyy, N. Desai, J. Mendling, and M. Weske, "Process compliance analysis based on behavioural profiles," *Inf. Syst.*, vol. 36, no. 7, pp. 1009–1025, Nov. 2011.

- [23] M. Weidlich, J. Mendling, and M. Weske, "Efficient consistency measurement based on behavioral profiles of process models," *IEEE Trans. Softw. Eng.*, vol. 37, no. 3, pp. 410–429, May/Jun. 2011.
- [24] W. van der Aalst, A. Adriansyah, and B. van Dongen, "Replaying history on process models for conformance checking and performance analysis," *Wires Data Mining Knowl. Discovery*, vol. 2, no. 2, pp. 182–192, 2012.
- [25] A. Rozinat and W. M. P. van der Aalst, "Conformance testing: Measuring the fit and appropriateness of event logs and process models," in *Proc. Int. Workshops, BPI, BPD, ENEL, BPRM, WSCOBPM, BPS, BPM*. Berlin, Germany: Springer, 2006, pp. 163–176.
- [26] J. De Weerd, M. De Backer, J. Vanthienen, and B. Baesens, "A robust F-measure for evaluating discovered process models," in *Proc. IEEE Symp. Comput. Intell. Data Mining (CIDM)*, Apr. 2011, pp. 148–155.



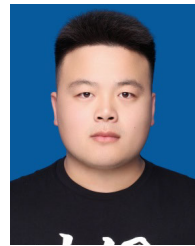
LI-LI WANG received the B.S. degree from the Anhui University of Science and Technology, China, in 2004, and the M.S. degree in software engineering from the Shandong University of Science and Technology, China, in 2007. She is currently pursuing the Ph.D. degree in information security engineering with the Anhui University of Science and Technology.

She is an Associate Professor with the Department of Computer Science and Engineering, School of Mathematics and Big Data, Anhui University of Science and Technology. Her research interests include Petri net and process mining.



XIAN-WEN FANG received the M.A. degree from the Shandong University of Science and Technology, China, in 2004, and the Ph.D. degree from the Key Laboratory of Service Computing, Tongji University, China, in 2011.

He is currently a Professor with the Department of Computer Science and Engineering, School of Mathematics and Big Data, Anhui University of Science and Technology, China. His research interests include Petri net, trustworthy software, and web services.



CHI-FENG SHAO received the B.S. and M.S. degrees from the Anhui University of Science and Technology, China, in 2016 and 2021, respectively.

He is currently a Teaching Assistant with the Computer Department, College of Information and Network Engineering, Anhui Science and Technology University. His research interests include Petri nets and process mining.



ESTHER ASARE received the B.S. degree in information technology from Valley View University, Ghana, Accra, in 2011, and the M.S. degree in health informatics from the University of Ghana, Legon, in 2015. She is currently pursuing the Ph.D. degree in information security engineering with the Anhui University of Science and Technology, Huainan, China.

Her research interests include the application process mining in the health industry, health IT, health policy, and health data security, among others. She became a member of the Ghana Health Informatics Association (GHIA) and HELINA (Health Informatics in Africa), in 2014.

...