IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 3, 2018, accepted December 2, 2018, date of publication January 14, 2019, date of current version June 7, 2019.

Digital Object Identifier 10.1109/ACCESS.2018.2885819

A Comprehensive Process Similarity Measure Based on Models and Logs

CHANGHONG ZHOU[1], CONG LIU[2], (Student Member, IEEE), QINGTIAN ZENG[3], ZEDONG LIN[2], AND HUA DUAN[4]

[1]College of Economics and Management, Shandong University of Science and Technology, Qingdao 266590, China
[2]College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China
[3]College of Electronic and Information Engineering, Shandong University of Science and Technology, Qingdao 266590, China
[4]College of Mathematics and System Science, Shandong University of Science and Technology, Qingdao 266590, China

Corresponding authors: Changhong Zhou (zhouchanghong@163.com), Qingtian Zeng (qtzeng@163.com), and Hua Duan (huaduan59@163.com)

This work was supported in part by the NSFC under Grant 61472229, Grant 61602278, Grant 61602279, and Grant 61702306), in part by the Science and Technology Development Fund of Shandong Province of China under Grant 2016ZDJS02A11, Grant ZR2017BF015, and Grant ZR2017MF027, in part by the Taishan Scholar Climbing Program of Shandong Province of China and SDUST Research Fund under Grant 2015TDJH102, in part by the Education Ministry Humanities and Social Science Project of China under Grant 16YJCZH012, Grant 16YJCZH154, Grant 16YJCZH041, and Grant 18YJAZH017, and in part by the Philosophy and Social Science Planning Project of Qingdao under Grant QDSKL1801141.

ABSTRACT Process similarity measure plays an important role in business process management and is usually considered as a versatile solution to fulfill the effective utilization of process models. Although many studies have worked on different notions of process similarity, most of them are not precise enough, as they simply compare processes with respect to the model structure features or the model behavior features separately. To address the problem, in this paper, we propose to measure the business process similarity by considering both process models and process logs. The process models are pre-defined descriptions of business processes, and the process logs can be considered as an objective observation of the actual process execution behavior. The combination of both can help to better character business processes. More specifically, two effective frameworks together with four novel approaches are presented. The former first constructs a weighted business process graph (WBPG) from the process model and the process log, and then computes the similarity of two corresponding WBPGs by using the weighted graph edit distance measure and the weighted node adjacent relation similarity measure. The latter first measures the similarity of process logs and the similarity of process models separately, and then merges the results. Finally, by experimental evaluation, we demonstrate the effectiveness and the applicability of the proposed approaches by comparing them with the start of the art.

INDEX TERMS Business process, process similarity, process model, process log.

I. INTRODUCTION

Business processes are important for modern enterprises and organizations. With rapid changes of the business environment, organizations need to be able to quickly and flexibly adjust their business processes to meet the new requirements. However, it is extremely complicated and time-consuming to construct business processes from scratch. Many advanced techniques, such as process recommendation, process query and process clustering, can facilitate organizations to reconstruct business processes in a rapid manner. These techniques are all based on the business process similarity [1]–[3].

Because of various application requirements, the definition of process similarity can be defined from different perspectives. For example, some existing works consider two processes similar if the textual labels of the elements in process models are similar [4]–[8]. Differently, some works measure the similarity by considering the process model topology [9]–[18] or the process model behavior [20]–[28]. Nevertheless, most of them are not precise enough, as they simply compare models with respect to the model structure features or the model behavior features separately. A comprehensive similarity measure is needed for a more precise measure.

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

In addition, the model behavior does not fully represent the actual process execution behavior. Typically, the process behavior is closely related to the organizations that execute it. This will lead to an interesting phenomenon: even if the same business process model is used, different organizations may observer totally/slightly different business behavior.

To clarify the problem, we introduce a simple online shopping scenario of three different electronic commerce companies, denoted as *ComA*, *ComB* and *ComC*. Fig. 1 shows their respective business process models represented as Petri nets [33], [34]. The legend in Fig. 1(d) shows the meaning of each transition, e.g., *B* refers to an activity named *pay by credit cards*. Obviously, these three business processes are similar according to the models. Considering the process models in Fig. 1(a)-(c) as an example, the processes in Fig. 1(b) and (c) represent two different payment selections of the process in Fig. 1 (a). Specifically, we have execution sequences *ABD* and *ACD* for Fig. 1(a), execution sequence *ABD* for Fig. 1(b) and execution sequence *ACD* for Fig. 1(c). Using the PTS-based similarity [25], we compute the similarity between Fig. 1(a) and Fig. 1(b) as 0.5, and the similarity between Fig. 1(a) and Fig. 1 (c) as 0.5, i.e., their similarities are identical.

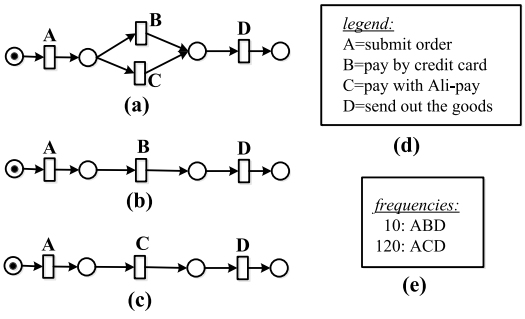


FIGURE 1. A motivating example.

The real execution behavior of a business process is recorded in the process log. The set of traces in Fig. 1 (e) is from the process log of the process in Fig. 1 (a). Each trace represents an execution sequence and the frequencies represent the number of times each trace occurs, e.g., trace *ABD* occurs 10 times. Based on the collected process log, we can see that trace *ACD* is more frequently executed than *ABD* in the business of *ComA*. Therefore, the business of *ComC* is more similar with the business of *ComA* than that of *ComB* if we take the behavior recorded in the log into account.

Therefore, we argue that the process similarity measure should also consider the real execution behavior in the log rather than only looking at the pre-defined process descriptions (e.g., models and textural descriptions). In this paper, we investigate two frameworks that measure process similarity by considering both process models and process logs.

The major contributions of this paper are as follows:

1) We aim to provide a comprehensive process similarity measurement including two frameworks and four approaches that considers both the process structure and the process behavior.

2) The comprehensive measurement uses the real executed log behavior, not the simulation behavior of the process model, to represent the process behavior. Therefore, the importance of different branches in the process models in different executed organizations can be distinguished.

3) We define the weighted graph edit distance by extending the traditional Graph Edit Distance idea according to the weighted business process graph features. And we apply the idea of EMD to the log similarity calculation and normalize it to get the log behavior pattern similarity.

4) We compare the proposed methods with the traditional process similarity traditional measures to demonstrate the effectiveness and the applicability by experiments.

The rest of this paper is organized as follows: Section II introduces some related work. Section III introduces some basic notations that will be used throughout the paper. In Section IV, we introduce the construction of Weighted Business Process Graph (*WBPG*) and two similarity measures based on *WBPGs*. Section V provides another two similarity measures. Section VI conducts experimental evaluation. Finally, Section VII concludes the paper.

II. RELATED WORK

Business process similarity can be measured from the following three different aspects of a process model: model textual similarity, model structural similarity and model behavioral similarity [25]. In this section, we summarize some of the related work. Afterwards, we point out the limitations of existing work.

A. PROCESS MODEL TEXTUAL SIMILARITY

The textual similarity mainly refers to the label textual information similarity of the elements contained in the process models. It is based on the fact that process models are composed of labeled nodes (task labels, event labels, etc.). This metric starts from calculating an optimal matching between the nodes in the process models by comparing their labels. A common technique is the consideration of (normalized) edit distances like the Levenshtein distance [4]. And other approaches in [5], [6] apply techniques from area of Natural Language Processing (*NLP*) in addition, thereby taking into account, for example, semantic information of node labels concerning synonyms, homonyms, antonyms, and so forth.

Based on this matching, a similarity score is calculated by taking into account the overall size of the models. Such labels are used for process similarity measure, i.e., the more similar labeled nodes they have, the more similar these processes are. Akkiraju and Ivan [7] measure similarity of process models solely based on the number of equally labeled activities. Minor *et al.* [8] suggest a measure that relates the number of nodes and edges to the overall number of nodes and edges in both models.

B. PROCESS MODEL STRUCTURAL SIMILARITY

The structural similarity mainly reflects the similarity of the process model topology which expresses the logical relationship between business activities. It depends on the relation of the relevant business data and the control-flow. Therefore, structure is the one of the important static attributes of the process model. The relevant aspects of this category arise from graph theory. And the general graph structured-based similarity between models can be quantified by, for example, the graph edit distance [9], [10] or the graph morphism detection [11] of two models, etc. Li *et al.* [12] define the graph edit distance between different Petri net process models and design the basic edit operations and similarity formula.

Alternatively, the construction of special graph-like representations, such as trees, are used to determine the similarity between such representations. In [13], [14], a process model is transformed into an ordered tree and the similarity of process models is measured on the base of tree edit distance. Attributed graphs are used to represent the process models and the process similarity is measured by considering both unit similarity and sequence similarity on optimal matching of weighted bipartite graphs in [15]. Graphs are compared considering sub-graph composition and a business process similarity factor is extracted in a modular process design [16]. The approaches introduced in [17] measure the distance between two process models by measuring their difference in terms of dependencies among activities. A block-structured process model is constructed based on a set of pre-defined blocks, i.e., sequences, branching, and loops with unique start and end nodes, in [18]. Since general graph-based algorithms do not consider any connectors (i.e., gateways), such connectors are often ignored.

C. PROCESS MODEL BEHAVIORAL SIMILARITY

Behavioral similarity emphases the execution semantics of business process models. This is usually expressed by a set of allowed execution traces of the process model. Such traces can be generated through simulation or during the actual execution of a process, and are usually stored in a process log. At present, most process behavioral similarity is obtained by measuring the similarity of simulated traces of process models. However, the computational complexity is extremely high when calculating the similarity of two process models based on the model simulation due to concurrent and loop structures in the process models [19]. To solve this problem, abstractions have been used. A typical application of abstraction is the transition adjacency relation (TAR for short) that considers pairs of activities that can be executed directly after each other [20]. TAR algorithm represents the behavior of a process model by transition adjacency relations and computes the similarity by the *Jaccard distance* of the two sets. Another abstraction is based on the weak order relations [21] which consider any pair of activities that can be executed after each other eventually. An extension of this abstraction is the so-called behavioral profiles, which distinguishes these relations by mutual exclusion, strict, and interleaving separately relations [22], [23]. In order to improve the effectiveness of BP, a new method is proposed in [24] for measuring the behavioral similarity between process models named TOR based on the occurrence relation among tasks. In addition, there are other methods that aim to tackle the infinite traces. For example, [25] defines three types of principal transition sequences and measures the similarity of each type separately. A behavioral process similarity algorithm is proposed in [26] based on complete firing sequences which are used to express model behavior. An approach named Transition-labeled Graph Edit Distance (TAGER) is introduced to calculate the similarity based on edit distance between coverability graphs in [27]. Liu *et al.* [28] propose a comprehensive approach to measure the process behavior similarity based on the so-called Extended Transition Relation set, ETR-set for short. Essentially, the ETR-set is an extended transition relation set consisting of direct causal transition relations, minimum concurrent transition relations and transitive causal transition relations.

D. SUMMARY OF EXISTING WORK

Because the textural similarity measures only consider the label set, it lacks lots of structural and behavioral information. Comparatively, the similarity measures based on topological structure and behavioral profiles have better performance and provide a more convincing result. However, existing works measure similarity by solely considering structural similarity or behavioral similarity but never combining both. Moreover, existing approaches of behavioral similarity are based on the process model and do not fully consider the actual execution behavior. Specifically, they do not consider the scenario that some parts of the process may be more important (more frequently executed) than others, i.e., there may be parts of the process model that are rarely activated while other parts are executed more often [29]. Obviously, this execution preference should be taken into account when measuring process behavioral similarity. Normally, these execution behavioral features are recorded in process logs. Therefore, we argue that the process similarity measure should combine the log behavior with the model structure. In this paper, we propose four different process similarity measure approaches by combining process models and process logs. These approaches provide a novel perspective to measure business processes in real-life environment.

III. PRELIMINARIES

Our work is based on process models and process logs. Process models are represented by business process graphs in order to deal with heterogeneous processes. We introduce the basic concepts of business process graph and process log in the following.

A. BUSINESS PROCESS GRAPH

A business process is a collection of related tasks that lead to a specified goal. Many modeling notations are available

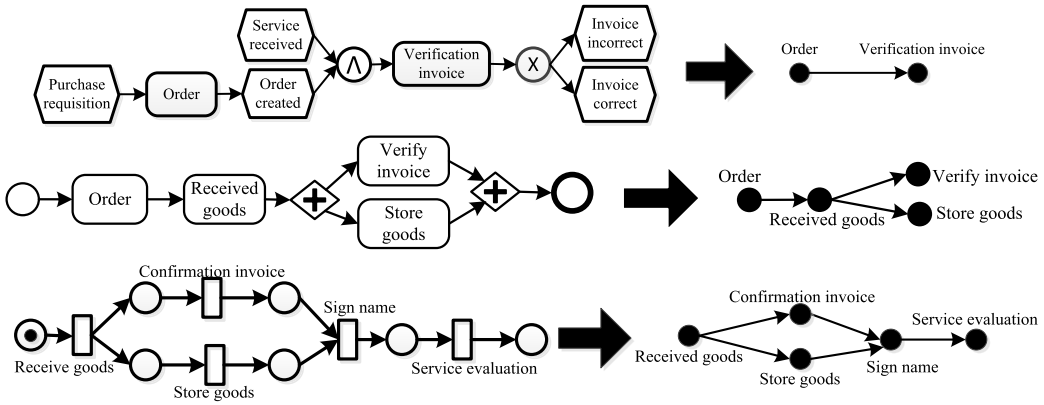


FIGURE 2. Three process models and their business process graphs.

to capture business processes, including Event-driven Process Chains (EPC) 30], UML Activity Diagrams [31], Business Process Modeling Notation (BPMN) [32] and Petri nets [33]–[37]. In this paper, we seek to abstract as much as possible from the specific notations to represent process models by using business process graphs. It is beneficial for measuring similarity of heterogeneous business process models.

Definition 1: A *business process graph* (*BPG*) is a 4-tuple $G = (N, E, \Gamma, \lambda)$, where (1) N is a set of nodes; (2) $E \subseteq N \times N$ is a set of edges; (3) Γ is a set of labels ; and (4) $\lambda : N \rightarrow \Gamma$ is a function that maps nodes to labels.

Definition 2: Let $G = (N, E, \Gamma, \lambda)$ be a *BPG* and $n, m \in N$ be two nodes. There is a path from n to m iff there exists a series of nodes $n_1, n_2, \ldots, n_k \in N$ such that $n_1 = n, n_k = m$ and for all $i \in 1, 2, \ldots, k-1$, $(n_i, n_{i+1}) \in E$, denoted as $n \Rightarrow m$. Then, $\{m | m \Rightarrow n\}$ is defined as the *Pre-sets* of n, denoted as n^{pre}, and $\{m | n \Rightarrow m\}$ is defined as the *Post-sets* of n, denoted as n^{post}.

Business process models represented by existing graphical notations can be easily transformed into *BPG*s. When transforming a process model to a *BPG*, we may discard certain types of nodes and focus on the task nodes only. According to [10], the main transformation rule contains two steps: (1) task nodes are identified and are represented as nodes in the *BPG* with the same labels; (2) if there exists a directed path from one task node to another task node and no other task nodes on this path, then we add an edge from the initial task node to the target task node in the corresponding *BPG*. Fig. 2 shows three process models in the form of EPC, BPMN and Petri net and their transformation to *BPG*s. The left column shows the original process models. The right column shows the corresponding *BPGs* after abstracting away some nodes (e.g. events, connectors/gateways, and places).

B. PROCESS LOG

A process log is defined as a set of cases where each case refers to an independent execution of the business process. A case consists of a sequence of events. For each event, it may have different attributes, e.g., activity name, timestamp, organization, resource, and etc. Note that we only consider the activity name attribute in this paper, i.e., each event refers to an activity name.

TABLE 1. A process log example.

Case ID	Event ID	ActivityName
c_1	e_1	a
	e_2	b
	e_3	c
	e_4	d
c_2	e_5	a
	e_6	c
	e_7	b
	e_8	d
c_3	e_9	a
	e_{10}	e
	e_{11}	d
c_4	e_{12}	a
	e_{13}	b
	e_{14}	c
	e_{15}	d
c_5	e_{16}	a
	e_{17}	b
	e_{18}	c
	e_{19}	d
c_6	e_{20}	a
	e_{21}	e
	e_{22}	d
c_7	e_{23}	a
	e_{24}	c
	e_{25}	b
	e_{26}	d
c_8	e_{27}	a
	e_{28}	e
	e_{29}	d
c_9	e_{30}	a
	e_{31}	e
	e_{32}	d

Table 1 gives a process log example which can be expressed as $L = \{c_1, c_2, \ldots, c_9\}$. The corresponding event

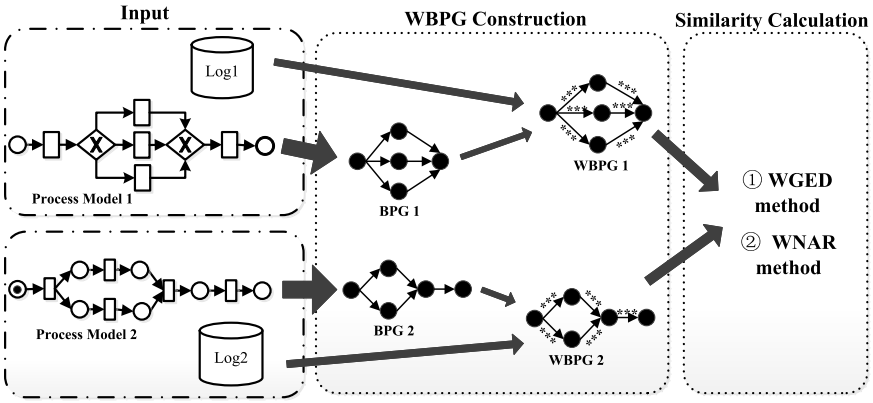


FIGURE 3. An overview of the WBPG-based similarity measure.

set is $\{e_1, e_2, \ldots, e_{32}\}$ and the activity set is $\{a, b, c, d, e\}$. The sequence that is made up of time-ordered events in one execution is called *case*. The resulting sequence by replacing the event with the activity name is called *trace*. The same trace may be occurred in different cases. Therefore, the process log can be defined as a multi-set of traces [38].

Definition 3: Let A be a set of activity names. A trace σ is defined as a sequence of activities, i.e. $\sigma \in A^*$. A process log L is a multi-set of sequences over set A.

For example, the process log in Table 1 contains nine traces. Let $\#_{frequency}(\sigma, L)$ represents the frequency of a trace σ in log L. Then for the example process log L in Table 1, we have $\#_{frequency}(<a, b, c, d>, L) = 3$, $\#_{frequency}(<a, c, b, d>, L) = 2$ and $\#_{frequency}(<a, e, d>, L) = 4$.

IV. WEIGHTED BUSINESS PROCESS GRAPH-BASED SIMILARITY MEASURE

This section presents a framework to measure the process similarity by combining both process model structure and log behavior based on the Weighted Business Process Graph (*WBPG*).

A. FRAMEWORK OF THE WBPG-BASED SIMILARITY MEASURE

The main idea of the *WBPG*-based similarity measure is to merge the structural information with the behavioral information in the log to calculate the process similarity. An overview of this approach is shown in the Fig. 3.

According to Fig. 3, we first construct a *WBPG* to merge the structural and the behavioral information. Then, two similarity measures, i.e., the Weighted Graph Edit Distance (*WGED*)–based approach and the Weighted Node Adjacent Relation sets (*WNAR*)-based approach, are proposed on top of the *WBPG*.

B. WEIGHTED BUSINESS PROCESS GRAPH

In the following, the definition of the weighted business process graph is formalized on the basis of *BPG*. Note that R denotes the real number set.

Definition 4: A *weighted business process graph* (*WBPG*) is defined as a 5-tuple $WG = (N, E, \Gamma, \lambda, f)$, where

1) N is a set of nodes;
2) $E \subseteq N \times N$ is a set of edges;
3) Γ is a set of labels;
4) $\lambda : N \rightarrow \Gamma$ is a function that maps nodes to labels;
5) $f : E \rightarrow R$ is a function that maps edges to real numbers which denote the weight.

The weight of an edge in *WBPG* is a normalized value and is represented as a real number. The *WBPG* is constructed by weighting the directed edges of the corresponding *BPG* based on the process log. In this way, behavioral information included in the process log can be incorporated into the process model.

To construct the *WBPG*, the main work is to traverse the process log on the basis of the process model expressed by *BPG*. Each edge is represented by an activity pair, i.e., $<a, b>$, and the number of sub-sequences that starts with activity a and ends with activity b in all traces can be counted after one traverse of the log. Then, the weight of $<a, b>$ is set as the ratio of the statistical sub-sequence frequency of all the trace frequencies in the log. Detailed computation process is organized in the Algorithm 1.

The time complexity of Algorithm 1 is $|E| * |L| * n$ where $|E|$ is the number of edges in the *BPG*, $|L|$ is the number of traces in the log, and n is the number of the activities in the trace.

Fig. 4 shows a Petri net model PM_1 and its corresponding *BPG*. Assume that we have the following process log L_1. It contains 12 traces and the *i-th* trace is $\sigma_i (1 \leq i \leq 12)$, i.e.,

$$\sigma_1 = <a, b, c, d, f, g, h>, \quad \sigma_2 = <a, b, c, e, f, g, h>,$$
$$\sigma_3 = <a, b, c, f, d, g, h>, \quad \sigma_4 = <a, b, c, f, e, g, h>,$$
$$\sigma_5 = <a, b, c, f, g, d, h>, \quad \sigma_6 = <a, b, c, f, g, e, h>,$$
$$\sigma_7 = <a, b, f, g, c, d, h>, \quad \sigma_8 = <a, b, f, g, c, e, h>,$$
$$\sigma_9 = <a, b, f, c, d, g, h>, \quad \sigma_{10} = <a, b, f, c, e, g, h>,$$
$$\sigma_{11} = <a, b, f, c, g, d, h>, \quad \sigma_{12} = <a, b, f, c, g, e, h>.$$

Algorithm 1 Weighted Business Process Graph Construction

Input: Business process graph $G = N, E, \Gamma, \lambda)$ and process log L
Output: Weighted Business Process Graph $WG = (N, E, \Gamma, \lambda, f)$
Let the frequency of the trace σ in L as $k(\sigma)$.
For each $e \in E$ // let a is the starting node and b is the ending node of e
$f(e) = 0$; $w = 0$;
For each $\sigma \in L$ // suppose $\sigma = t_1 t_2 \ldots t_{n-1} t_n$
$Flag(t_1) = Flag(t_2) = \ldots = Flag(t_n) = 0$;
For i = 1 to n
if ($t_i = a$) and ($Flag(t_i) = 0$) then
For j = i to n
if ($t_j = b$) and ($b \in t_i^{post}$) and ($Flag(t_j) = 0$) then // trace σ passed the edge e
$Flag(t_i) = Flag(t_j) = 1$;
$f(e) = f(e) + k(\sigma)$; // to make the weight accumulation for the edge e
Endif
Endfor
Endif
Endfor
$w = w + max(f(e), k(\sigma))$;
Endfor
$f(e) = f(e)/w$; // to normalize the weights of each edge
Endfor

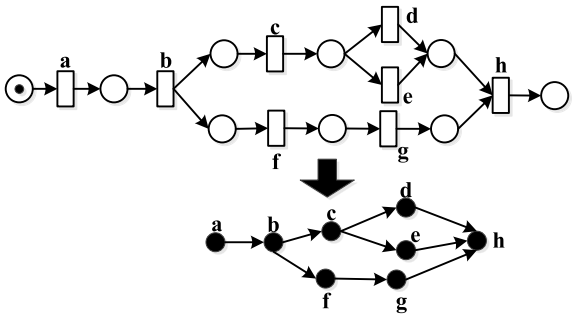


FIGURE 4. A petri net process model PM_1 and corresponding *BPG*.

The frequency of each trace is given as follows:

$$\#_{frequency}(\sigma_1, L_1) = 50, \quad \#_{frequency}(\sigma_2, L_1) = 15,$$
$$\#_{frequency}(\sigma_3, L_1) = 60, \quad \#_{frequency}(\sigma_4, L_1) = 5,$$
$$\#_{frequency}(\sigma_5, L_1) = 50, \quad \#_{frequency}(\sigma_6, L_1) = 10,$$
$$\#_{frequency}(\sigma_7, L_1) = 80, \quad \#_{frequency}(\sigma_8, L_1) = 2,$$
$$\#_{frequency}(\sigma_9, L_1) = 60, \quad \#_{frequency}(\sigma_{10}, L_1) = 3,$$
$$\#_{frequency}(\sigma_{11}, L_1) = 100, \quad \#_{frequency}(\sigma_{12}, L_1) = 5.$$

Take the directed edge $< c, d >$ of the *BPG* in Fig. 4 as an example. For trace σ_1, there is activity d appearing after activity c(not necessarily with immediate presence). According to Algorithm 1, the frequency of the edge $< c, d >$ in trace σ_1 is 50, i.e. $\#_{frequency}(<c, d>, \sigma_1) = 50$. Similarly,

$$\#_{frequency}(<c, d>, \sigma_2) = 0, \quad \#_{frequency}(<c, d>, \sigma_3) = 60,$$
$$\#_{frequency}(<c, d>, \sigma_4) = 0, \quad \#_{frequency}(<c, d>, \sigma_5) = 50,$$
$$\#_{frequency}(<c, d>, \sigma_6) = 0, \quad \#_{frequency}(<c, d>, \sigma_7) = 80,$$
$$\#_{frequency}(<c, d>, \sigma_8) = 0, \quad \#_{frequency}(<c, d>, \sigma_9) = 60,$$
$$\#_{frequency}(<c, d>, \sigma_{10}) = 0, \quad \#_{frequency}(<c, d>, \sigma_{11}) = 100,$$
$$\#_{frequency}(<c, d>, \sigma_{12}) = 0.$$

Take the directed edge $< c, d >$ of the *BPG* in Fig. 4 as an example. For trace σ_1, there is activity d appearing after activity c(not necessarily with immediate presence). According to Algorithm 1, the frequency of the edge $< c, d >$ in trace σ_1 is 50, i.e. $\#_{frequency}(<c, d>, \sigma_1) = 50$. Similarly,

$$\#_{frequency}(<c, d>, \sigma_2) = 0, \quad \#_{frequency}(<c, d>, \sigma_3) = 60,$$
$$\#_{frequency}(<c, d>, \sigma_4) = 0, \quad \#_{frequency}(<c, d>, \sigma_5) = 50,$$
$$\#_{frequency}(<c, d>, \sigma_6) = 0, \quad \#_{frequency}(<c, d>, \sigma_7) = 80,$$
$$\#_{frequency}(<c, d>, \sigma_8) = 0, \quad \#_{frequency}(<c, d>, \sigma_9) = 60,$$
$$\#_{frequency}(<c, d>, \sigma_{10}) = 0, \quad \#_{frequency}(<c, d>, \sigma_{11}) = 100,$$
$$\#_{frequency}(<c, d>, \sigma_{12}) = 0.$$

Then, the weight of edge $< c, d >$ can be computed as:

$$f(<c, d>) = \frac{50+60+50+80+60+100}{50+15+60+5+50+10+80+2+60+3+100+5} = 0.91$$

The weight of other edges can be computed in the same way and the obtained weighted business process graph WG_1 is shown in Fig. 5.

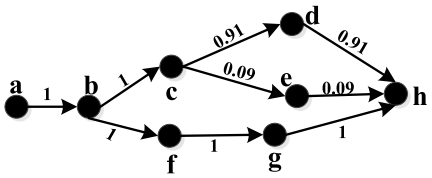


FIGURE 5. The constructed *WBPG* WG_1.

The main problem is the drop of the connector/gateway nodes so that the branch relationship can't be distinguished between choice and concurrent when we transform a process model to process graph. By weighting for the *BPG*, it is possible to distinct the different branch relationship based on the weight of edges.

1) Two edges have the same weight that are composed of one of the two task nodes with a concurrence of relationship and their nearest public predecessor node. For example, the node c and f have the concurrent relationship in the petri net model of Fig. 4. So the edge $<b, c>$ and the edge $<b, f>$ have the same weight of 1. In addition, the weight of the edge $<a, b>$ before the concurrent of c and f occurs is also 1. That is to say, all three values are identical.

2) Two edges have usually different weights that are composed of one of the two task nodes with a choice of relationship and their nearest public predecessor node. For example, the node d and the node e have the choice of relationship, i.e. the task node e cannot be executed at the same time as the task node d is running, and vice versa. From Fig.5, we can see that the weight of the edge $<c, d>$ is 0.91 while the weight of the edge $<c, e>$ is 0.09. And They are all little than the weight of the edge $<b, c>$

C. WEIGHTED GRAPH EDIT DISTANCE

To compare two *WBPGs*, we define a metric based on the notion of *WGED*. The *WGED* extends the definition of the graph edit distance in [10]. It is the minimal cost of transforming one graph into the other. Transformations are captured as sequences of elementary transformation operations including node substitution, node insertion/deletion and edge insertion/deletion. Each elementary operation has a cost, which is given by a cost function. The more similar two graphs are, the smaller the graph edit distance they have, i.e. the smaller the transformation cost is. However, different edges may have different weight values. Therefore, the corresponding operational cost should have a direct correlation with the weight values of the edges. In general, the greater the weight change of an edge, the higher the cost to operate it.

Definition 5: Let $WG_1 = (N_1, E_1, \Gamma_1, \lambda_1, f_1)$ and $WG_2 = (N_2, E_2, \Gamma_2, \lambda_2 f_2)$ be two *WBPGs*. Let $M : N_1 N_2$ be a partial injective mapping that maps nodes in WG_1 to nodes in WG_2. Let $dom(M) = \{n_1 | (n_1, n_2) \in M\}$ be the domain of M and $cod(M) = \{n_2 | (n_1, n_2) \in M\}$ be the co-domain of M. We define the following basic operations:

1) Given a node $n \in N_1 \cup N_2$, n is substituted if $n \in dom(M)$ or $n \in cod(M)$. *subn* represents the set of all substituted nodes.

2) A node $n \in N_1$ is deleted from WG_1(or inserted to WG_2) if $n \notin subn$. A node that is deleted from WG_2(or inserted to WG_1) is defined in the same way. *skipn* represents the set of all nodes deleted and inserted.

3) Let $(n_1, m_1) \in E_1$ be an edge of WG_1. (n_1, m_1) is deleted from WG_1(or inserted to WG_2) if there does not exist a mapping M such that $(n_1, n_2) \in M$ and $(m_1, m_2) \in M$ and $(n_2, m_2) \in E_2$. Edges that are deleted from WG_2(or inserted to WG_1) are defined in the same way. *skipe* represents the set of all inserted and deleted edges.

4) Let $(n_1, m_1) \in E_1$ be an edge of WG_1. (n_1, m_1) is substituted if it is not inserted or deleted. *sube* represents the set of all substituted edges, i.e., $sube = (E_1 \cup E_2) - skipe$.

For the edit distance operation, textual similarity between labels of two nodes are required. The textural similarity is defined on the basis of the string similarity as defined in the following.

Definition 6: Let s and t be two strings and let $|x|$ be the length of string x. The *string edit distance* of s and t, denoted $sed(s, t)$ is the minimal number of atomic string operations needed to transform s to t or vice versa. The atomic string operations include: inserting a character, deleting a character and substituting a character by another one.

Definition 7: Let $WG_1 = (N_1, E_1, \Gamma_1, \lambda_1, f_1)$ and $WG_2 = (N_2, E_2, \Gamma_2, \lambda_2, f_2)$ be two *WBPGs*, and $n_1 \in N_1$ and $n_2 \in N_2$ are two nodes. The similarity of n_1 and n_2 is computed as follows:

$$Nsim(n_1, n_2) = 1.0 - \frac{sed(\lambda_1(n_1), \lambda_2(n_2))}{max(|\lambda_1(n_1)|, |\lambda_2(n_2)|)}$$

Then, we define the weighted graph edit distance as follows.

Definition 8: Let $WG_1 = (N_1, E_1, \Gamma_1, \lambda_1, f_1)$ and $WG_2 = (N_2, E_2, \Gamma_2, \lambda_2, f_2)$ be two *WBPGs*. Let $M : N_1 N_2$ be a partial injective mapping that maps nodes in WG_1 to nodes in WG_2. Let $dom(M) = \{n_1 | (n_1, n_2) \in M\}$ be the domain of M and $cod(M) = \{n_2 | (n_1, n_2) \in M\}$ be the co-domain of M. The weighted graph edit distance that is based on the mapping M is computed as follows:

$$WGED_M(WG_1, WG_2) = \| skipn \| + \| skipe \| + \| sube \| + \| subn \|$$

where:

$\|skipn\|$ is the operational cost of node insertion and node deletion. It is defined as the total number of the inserted and deleted nodes, i.e., $\| skipn \| = |skipn|$;

$\|skipe\|$ is the operational cost of edge insertion and edge deletion. It is defined as the sum of the weights of the inserted and deleted edges, i.e., $\| skipe \| = \sum_{e \in skipe \wedge e \in E_1} f_1(e) + \sum_{e \in skipe \wedge e \in E_2} f_2(e)$;

$\|sube\|$ is the operational cost of edge substitution. It is defined as the sum of the absolute values of the difference between the weights of the corresponding substituted edges, i.e., $\|sube\| = \sum_{e \in sube} |f_1(e) - f_2(e)|$; and

$\|subn\|$ is the operational cost of n ode substitution. It can be computed based on the Definitions 6-7, i.e., $\|subn\| = 2 \times \sum_{(n1,n2) \in M} (1\text{-}Nsim(n_1, n_2))$.

The *WGED* of the two *WBPGs* can be computed as the minimal possible distance based on mapping M:

$$WGED(WG_1, WG_2) = \min_M WGED_M(WG_1, WG_2)$$

Let *subn*, *skipn*, *skipe* and *sube* be the sets of substituted nodes, inserted/deleted nodes, inserted/deleted edges, and substituted edges and $0 \leq wsubn \leq 1$, $0 \leq wskipn \leq 1$, $0 \leq wskipe \leq 1$, $0 \leq wsube \leq 1$ be the weights that we assign to nodes substitution, nodes insertion/deletion, edges insertion/deletion, and edges substitution. We define *fskipn*, *fskipe*,*fsubn*, and *fsube* as follows:

$$fskipn = \frac{|skipn|}{|N_1| + |N_2|},$$

$$fsubn = \frac{2.0 \times \sum_{(n,m) \in M} (1.0 - Nsim(n, m))}{|subn|}$$

$$fskipe = \frac{\sum_{e \in skipe \wedge e \in E_1} f_1(e) + \sum_{e \in skipe \wedge e \in E_2} f_2(e)}{\sum_{e \in E_1} f_1(e) + \sum_{e \in E_2} f_2(e)},$$

$$fsube = \frac{\sum_{e \in sube} |f_1(e) - f_2(e)|}{\sum_{e \in sube} \max(f_1(e), f_2(e))}.$$

where *fskipn* represents the fraction of inserted/deleted nodes, *fskipe* represents the fraction of inserted/deleted edges, *fsubn* represents the average distance of substituted nodes, and *fsube* represents the average changes in weight value of substituted edges.

The weighted graph edit similarity based on mapping M is defined as shown at the bottom of this page.

Take the process model PM_1 in Fig. 4 and its weighted business process graph WG_1 in Fig. 5 as an example. The other model PM_2 in Fig. 6 (a) has the process log L_2 which contains 2 traces. The *i-th* trace is σ_I', i.e., $\sigma_1' = <a, b, c, d, h>, \sigma_2' = <a, b, c, g, h>$. The frequency of each trace is given as follows: $\#_{frequency}(\sigma_1', L_2) = 30$, $\#_{frequency}(\sigma_1', L_2) = 70$. According to Algorithm 1, the weighted business process graph WG_2 of PM_2 can be constructed as shown in Fig. 6 (b).

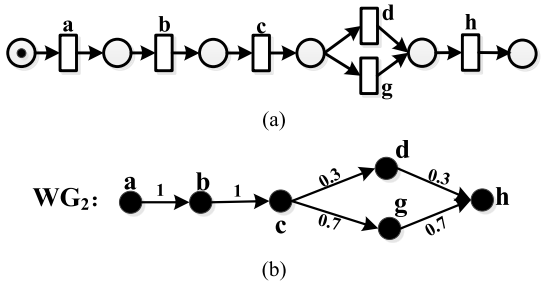


FIGURE 6. Process model PM_2 and corresponding *WBPG* WG_2.

The substituted nodes can be neglected because the same activity labels lead to the zero of the *fsubn*. Therefore, we only consider skipped nodes, skipped edges, and substituted edges. Using the weights $wskipn = 0.2$, $wskipe = 0.6$, $wsube = 0.6$ and $wsubn = 0.7$, the similarity is computed as follows:

$$\begin{aligned}
&\| skipn \| = |skipn| = 2;\\
&\| skipe \| = 1 + 1 + 0.09 + 0.09 + 0.7 = 2.88;\\
&\| sube \| = |1 - 1| + |1 - 1| + |0.91 - 0.3|\\
&\qquad + |0.91 - 0.3| + |1 - 0.7| = 1.52;\\
&fskipn = \frac{2}{8+6} \approx 0.143; \quad fskipe = \frac{2.88}{7+4} \approx 0.262;\\
&fsube = \frac{1.52}{1+1+0.91+0.91+1} \approx 0.315.\\
&sim_{WGED}(WG_1, WG_2) = 1.0\\
&\quad - \frac{0.2 \times 0.143 + 0.6 \times 0.262 + 0.6 \times 0.315 + 0.7 \times 0}{0.2 + 0.6 + 0.6 + 0.7}\\
&\quad \approx 0.8215
\end{aligned}$$

D. WEIGHTED NODE ADJACENT RELATION SIMILARITY

In the *BPG*, an edge represents the adjacent relation of two relevant nodes. To compare two graphs, we consider the percentages of their common edges. However, for the *WBPG*, the importance of node adjacent relation is denoted by the weight of the edge connected by the corresponding nodes. In this way, only the percentage of the common edge number is not enough to represent the similarity of two weighted graphs. In this section, we use the Weighted Node Adjacent Relation (short for *WNAR*) to measure the similarity of two weighted business process graphs.

Definition 9: Let $WG = (N, E, \Gamma, \lambda, f)$ be a *WBPG*. If there exists two nodes $n, m \in N$ such that $e = <n, m> \in E$, then a tuple $<n, m>$ with its weight $f(e)$ is called a *WNAR*. For a given *WBPG*, all *WNAR*s of a *WBPG* form *WNAR set*, denoted as $WNARs = \{f(e)<n, m> | e = <n, m> \in E\}$.

For example, the *WNAR set* of the *WBPG* in Fig. 5 is $\{1<a, b>, 1<b, c>, 1<b, f>, 0.91<c, d>, 0.09<c, e>, 1<f, g>, 0.91<d, h>, 0.09<e, h>, 1<g, h>\}$. The *WNAR* set represents the node adjacent relations and their importance. Obviously, the *WNAR* set can be regarded as a multi-set. The repeatability of the multi-sets is the weight. So the operations on the sets should be based on multi-sets.

Definition 10: Let $WG_1 = (N_1, E_1, \Gamma_1, \lambda_1, f_1)$ and $WG_2 = (N_2, E_2, \Gamma_2, \lambda_2, f_2)$ be two *WBPGs*. Let *sube* and *skipe* be the sets of substituted edges and inserted/deleted edges separately as defined in Definitions 5 and 8. The *WNARs* of the WG_1 and WG_2 are named $WNARs_1$ and $WNARs_2$ respectively. The intersection and union operations of $WNARs_1$ and $WNARs_2$ are defined as follows:

$$\begin{aligned}
&WNARs_1 \Cap WNARs_2 = \{\omega_i e_i | \omega_i = min(f_1(e_i), f_2(e_i)), e_i \in sube\}\\
&WNARs_1 \Cup WNARs_2 = \{\omega_i e_i | \omega_i = max(f_1(e_i), f_2(e_i)), e_i \in sube\}\\
&\qquad \cup \{f_1(e_i) e_i | e_i \in skipe \wedge e_i \in WG_1\}\\
&\qquad \cup \{f_2(e_i) e_i | e_i \in skipe \wedge e_i \in WG_2\}
\end{aligned}$$

The *WNAR* sets similarity of WG_1 and WG_2 is defined as follows:

$$sim(WG_1, WG_2) = \frac{|WNARs_1 \Cap WNARs_2|}{|WNARs_1 \Cup WNARs_2|}$$

Consider for example WG_1 in Fig. 5 and WG_2 in Fig. 6. The *WNAR* sets of WG_1 and WG_2 are:

$WNARs_1 = \{1<a, b>, 1<b, c>, 1<b, f>, 0.91<c, d>, 0.09<c, e>, 1<f, g>, 0.91<d, h>, 0.09<e, h>, 1<g, h>\}$, and $WNARs_2 = \{1<a, b>, 1<b, c>, 0.3<c, d>, 0.7<c, g>, 0.3<d, h>, 0.7<g, h>\}$

Then, the similarity of WG_1 and WG_2 is computed as:

$$\begin{aligned}
&sim_{WNAR}(WG_1, WG_2)\\
&= \frac{1 + 1 + 0.3 + 0.3 + 0.7}{1 + 1 + 1 + 0.91 + 0.09 + 0.7 + 1 + 0.91 + 0.09 + 1}\\
&\approx 0.4286
\end{aligned}$$

$$sim_{WGED}(WG_1, WG_2) = 1.0 - \frac{wskipn \times fskipn + wskipe \times fskipe + wsubn \times fsubn + wsube \times fsube}{wskipn + wskipe + wsubn + wsube}$$

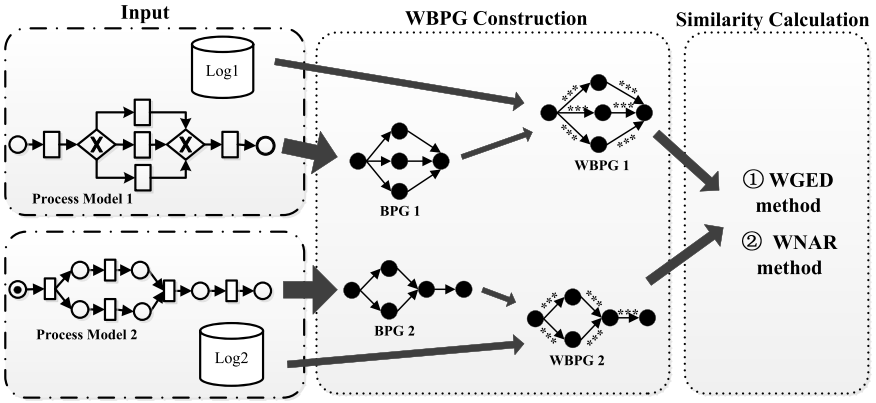


FIGURE 7. An overview of the *LBP*-based similarity measure.

V. LOG BEHAVIOR PATTERN-BASED SIMILARITY MEASURE

In this section, we propose another approach to measure the similarity of business processes based on the log behavior and model structure. The process log similarity is computed as the log behavior pattern (*LBP*) similarity.

A. FRAMEWORK OF THE LBP-BASED SIMILARITY MEASURE

Fig. 7 depicts an overview of the *LBP*-based framework. Compared to the *WBPG*-based similarity measure, the approach computes the *BPG* structural similarity and the log behavioral similarity separately. Based on this framework, a process model is first translated to a *BPG* and then existing graph similarity measures are used to calculate the structural similarity. Here, the Node Adjacent Relation (*NAR*) measure and the Graph Edit Distance (*GED*) measure are used. Then, for the similarity calculation of the logs, the Earth Mover's Distance (*EMD*) [39] is used to measure the Log Behavior Pattern (*LBP*) similarity. Finally, the similarity result can be obtained by the weighted merge of both the model structural similarity and the log similarity.

B. LBP SIMILARITY

In this section, we consider the similarity of two process logs. The similarity of two process logs is essentially the similarity between the two multi-sets of traces (or sequences).

Definition 11: Given a process log L, the log behavior pattern is composed of the set of two tuples (σ, β) where σ is a trace in L and β is the frequency of the trace σ in L, i.e., $\beta = \#_{frequency}(\sigma, L)$. For example, the log behavior pattern of the process log in Table 1 is described as: $\{(< a, b, c, d >, 3), (<a, c, b, d >, 2), (< a, e, d >, 4)\}$.

The process log similarity is measured by computing the similarity of the log behavior pattern. To calculate the similarity of log behavior patterns, we need to first calculate the log behavior pattern distance. Obviously, the log behavior pattern mainly contains the executed traces (sequences) of the process and their corresponding frequencies. The distance of two traces (sequences) is defined as follows.

Definition 12: Given two sequences S_1 and S_2, the distance between them is defined as follows:

$$D_{seq}(S_1, S_2) = 1 - \frac{|lcs(S_1, S_2)|}{|S_1| + |S_2| - |lcs(S_1, S_2)|} \tag{1}$$

where $|S|$ is the length of S and lcs (S_1, S_2) is the longest common subsequence of S_1 and S_2.

An intuition is that the longer the common subsequence of two traces, the more similar these two traces are.

Theory 1: The sequence distance is a metric.

Proof: Let sed (S_1, S_2) be the string edit distance of two sequences S_1 and S_2. According to the Equation (1), we have

$$\begin{aligned} D_{seq}(S_1, S_2) &= 1 - \frac{|lcs(S_1, S_2)|}{|S_1| + |S_2| - |lcs(S_1, S_2)|} \\ &= \frac{|S_1| + |S_2| - 2|lcs(S_1, S_2)|}{|S_1| + |S_2| - |lcs(S_1, S_2)|} \\ &= \frac{2|S_1| + 2|S_2| - 4|lcs(S_1, S_2)|}{2|S_1| + 2|S_2| - 2|lcs(S_1, S_2)|} \end{aligned} \tag{2}$$

According to the conclusion in [37], we have

$$sed(S_1, S_2) = |S_1| + |S_2| - 2|lcs(S_1, S_2)| \tag{3}$$

Based on Equations (2)-(3), we have:

$$D_{seq}(S_1, S_2) = \frac{2sed(S_1, S_2)}{|S_1| + |S_2| + sed(S_1, S_2)} \tag{4}$$

Equation (4) has been proved to be a metric and is a normalized edit distance in [40]. Therefore, the sequence distance $D_{seq}(S_1, S_2)$ is a metric.

As known in [39], the *EMD* naturally extends the notion of distance between different elements to the distance between sets of elements. The distance of two multi-sets, named as the log behavior pattern distance, is defined as follows.

Definition 13: Let $P = \{(p_1, w_{p1}), \ldots, (p_m, w_{pm})\}$ and $Q = \{(q_1, w_{q1}), \ldots, (q_m, w_{qm})\}$ be the behavior patterns of two process logs L_1 and L_2 respectively. f_{ij} is the conversion cost from the execution sequence p_i to q_j. And $D_{seq}(p_i, q_j)$ is the

sequence distance between p_i and q_j. Then, the log behavior pattern distance between P and Q is defined as:

$$D_{pat}(P, Q) = \min \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} D_{seq}(p_i, q_j)$$

w.r.t.:

$$\begin{cases} \sum_{j=1}^{n} f_{ij} = \frac{w_{pi}}{\sum_{i=1}^{m} w_{pi}}, & 1 \le i \le m,\ 1 \le j \le n; \\ \sum_{i=1}^{m} f_{ij} = \frac{w_{qj}}{\sum_{j=1}^{n} w_{qj}}, & 1 \le i \le m,\ 1 \le j \le n; \\ f_{ij} \ge 0, 1 \le i \le m,\ 1 \le j \le n; \end{cases}$$

Theory 2: The log behavior pattern distance is a metric.

Proof: The log behavior pattern distance is a special case of *EMD*. As proved in [39], if the base distance is a metric and the amount of two distributions is same, then the *EMD* is a metric. For the log behavior pattern distance, the sequence distance is the base distance and the sequence distance has been proved to be a metric in Theory 1. Therefore, the log behavior pattern distance defined from *EMD* is also a metric.

Therefore, the *LBP* similarity of L_1 and L_2, is defined as:

$$sim_{LBP}(L_1, L_2) = 1\text{-}D_{pat}(P, Q)$$

Take the process log L_1 of PM_1 in Fig. 4 and process log L_2 of PM_2 in Fig. 6 as an example. The *LBP* similarity value of L_1 and L_2 can be calculated as $sim_{LBP}(L_1, L_2) = 0.71$.

C. NAR-LBP MEASURE

Similar to the *WNAR*, Node Adjacent Relation (*NAR*) is used to compute the similarity between two process models. The *NAR*-based structural similarity is defined as the proportion of common edges between two *BPG*s.

Definition 14: Let $G = (N, E, \Gamma, \lambda)$ be a *BPG*. If there exists two nodes $n, m \in N$ such that $e = <n, m> \in E$, then a tuple $<n, m>$ is a *NAR*. For a given *BPG*, all *NAR*s of the business process graph form the *NAR set*, denoted as $NARs = \{<n, m> | e = <n, m> \in E\}$.

Definition 15: Given two business process graphs G_1 and G_2, the node adjacent relation sets are named by $NARs_1$ and $NARs_2$. Then, the similarity between *NAR sets* of G_1 and G_2 is defined as:

$$sim_{NAR}(G_1, G_2) = \frac{|NARs_1 \cap NARs_2|}{|NARs_1 \cup NARs_2|}$$

Considering PM_1 in Fig. 4 and PM_2 in Fig. 6 as an example, the corresponding *BPGs* is G_1 and G_2 in Fig. 8. The *NAR sets* of G_1 and G_2 are defined as follows:

$$\begin{aligned} &NARs_1 \\ &= \{<a, b>, <b, c>, <b, f>, <c, d>, <c, e>, \\ &\quad <f, g>, <d, h>, <e, h>, <g, h>\}; \\ &NARs_2 \\ &= \{<a, b>, <b, c>, <c, d>, <c, g>, <d, h>, \\ &\quad <g, h>\}; \\ &NARs_1 \cap NARs_2 \\ &= \{<a, b>, <b, c>, <c, d>, <d, h>, <g, h>\}; \end{aligned}$$

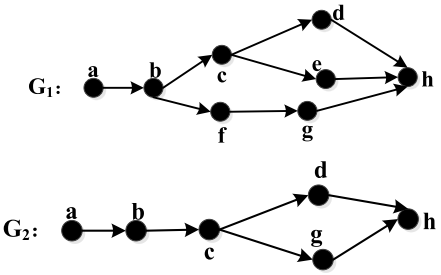


FIGURE 8. The corresponding *BPGs* of PM_1 and PM_2.

$$\begin{aligned} &NARs_1 \cup NARs_2 \\ &= \{<a, b>, <b, c>, <b, f>, <c, d>, <c, e>, \\ &\quad <c, g>, <f, g>, <d, h>, <e, h>, <g, h>\}. \end{aligned}$$

Then, the similarity between G_1 and G_2 is computed as:

$$sim_{NAR}(G_1, G_2) = \frac{5}{10} = 0.5$$

Definition 16: Let L_1 and L_2 be the process logs of two process models P_1 and P_2, G_1 and G_2 be the *BPG* of P_1 and P_2. The *NAR-LBP* similarity of P_1 and P_2 is defined as:

$$sim_{NAR-LBP}(P_1, P_2) = \alpha \times sim_{NAR}(G_1, G_2) + \beta \times sim_{LBP}(L_1, L_2)$$

where α and β are two coefficients and $\alpha + \beta = 1$.

Assume that $\alpha = 0.5$ and $\beta = 0.5$, the similarity of process model PM_1 in Fig. 4 and process model PM_2 in Fig. 6 based on the *NAR-LBP* measure is computed as follows:

$$\begin{aligned} &sim_{NAR\_LBP}(PM_1, PM_2) \\ &\quad = 0.5 \times sim_{NAR}(G_1, G_2) + 0.5 \times sim_{LBP}(L_1, L_2) \\ &\quad = 0.5 \times 0.5 + 0.5 \times 0.71 = 0.605 \end{aligned}$$

D. GED-LBP MEASURE

The Graph Edit Distance (*GED*) [10] is defined as the minimal cost of transforming one graph into the other by node substitution, node insertion/deletion, and edge insertion/deletion. For two business processes, the more similar the two corresponding business process graphs are, the smaller the graph edit distance between the two *BPG*s is. In addition, the cost of substitution operation can be determined based on the string edit distance and node similarity as mentioned in Definitions 6-7.

Definition 17: Let $G_1 = (N_1, E_1, \Gamma_1, \lambda_1)$ and $G_2 = (N_2, E_2, \Gamma_2, \lambda_2)$ be two *BPGs*. Let $M : N_1 N_2$ be a partial injective function that maps nodes in G_1 to nodes in G_2. Let $dom(M) = \{n_1 | (n_1, n_2) \in M\}$ be the domain of M and $cod(M) = \{n_2 | (n_1, n_2) \in M\}$ be the co-domain of M. Operations including *subn*, *skipn*, *skipe* and *sube* are defined in the same way as explained in Definition 5. The graph edit distance that is based on mapping M is defined as follows:

$$GED_M(G_1, G_2) = |skipn| + |skipe| + 2 \times \sum_{(n,m) \in M} (1 - Nsim(n, m))$$

The graph edit distance of the two business process graphs is the minimal possible distance based on mapping M. Let $0 \leq wskipn \leq 1$, $0 \leq wskipe \leq 1$, $0 \leq wsubn \leq 1$ be the weights that we assign to the inserted/deleted nodes, inserted/deleted edges and substituted nodes. The fraction of inserted/deleted nodes, denoted as *fskipn*, the fraction of inserted/deleted edges, denoted as *fskipe* and the average distance of substituted nodes, denoted as *fsubn*, are defined as follows:

$$fskipn = \frac{|skipn|}{|N_1| + |N_2|}, \quad fskipe = \frac{|skipe|}{|E_1| + |E_2|},$$
$$fsubn = \frac{2.0 \times \sum_{(n,m)\in M} (1.0 - Nsim(n, m))}{|subn|}.$$

Then, the graph edit similarity based on mapping M is computed as follows:

$$sim_{GED}(G_1, G_2) = 1.0 - \frac{wskipn \times fskipn + wskipe \times fskipe + wsubn \times fsubn}{wskipn + wskipe + wsubn}$$

Although there are four types of edit operations, only skipped nodes, skipped edges, substituted nodes are considered. The substituted edges represent the same edges between two business process graphs. If the fraction of the substituted edges is considered, it will lead to the graph edit similarity of the same two graphs smaller than 1. This does not conform to normal cognition.

Consider for example G_1 and G_2 in Fig. 8. Assume that the mapping is constructed between the nodes by the same activity name. Obviously, the *fsubn* is zero because the node similarity of the same name is 1. Transforming G_1 to G_2 can be done by deleting two nodes, deleting four edges and inserting one edge. If we set the weights as $wkipn = 0.2$, $wskipe = 0.6$ and $wsubn = 0.7$, the graph edit similarity of the two graphs in Fig. 8 is computed as follows:

$$|skipn| = 2, \quad |skipe| = 5, \ |subn| = 12,$$
$$fskipn = \frac{2}{8+6} \approx 0.143, \quad fskipe = \frac{5}{9+6} \approx 0.333,$$
$$fsubn = \frac{2.0 \times \sum_{(n,m)\in M} (1.0 - 1.0)}{12} = 0,$$
$$sim_{GED}(G_1, G_2) = 1.0 - \frac{0.2 \times 0.143 + 0.6 \times 0.333 + 0.7 \times 0}{0.2 + 0.6 + 0.7} \approx 0.8477.$$

Definition 18: Given two business process models P_1 and P_2 and their process logs L_1 and L_2. Let G_1 is the business process graph of P_1 and G_2 is the business process graph of P_2. The *GED-LBP* similarity of P_1 and P_2 is defined as:

$$sim_{GED-LBP}(P_1, P_2) = \alpha \times sim_{GED}(G_1, G_2) + \beta \times sim_{LBP}(L_1, L_2)$$

where α and β are two coefficients satisfying $\alpha + \beta = 1$.

Assume that $\alpha = 0.5$ and $\beta = 0.5$, the similarity of process model PM_1 in Fig. 4 and process model PM_2 in Fig. 6 based on the *GED-LBP* measure is computed as:

$$sim_{GED\_LBP}(PM_1, PM_2) = 0.5 \times sim_{GED}(G_1, G_2) + 0.5 \times sim_{LBP}(L_1, L_2) = 0.5 \times 0.8477 + 0.5 \times 0.71 \approx 0.779$$

VI. EXPERIMENTAL SETTING AND ANALYSIS

This section performs a comprehensive set of experiments to evaluate the proposed approaches. The experimental setting, experimental results and discussions are presented as follows.

A. EXPERIMENTAL SETTING

To illustrate the effectiveness of the proposed approaches, we first construct a group of artificial process model variants that have similar structure. Then, we generate process logs for these process model variants through simulation.

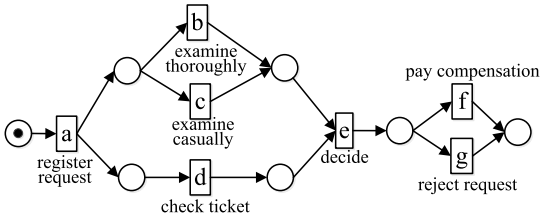


FIGURE 9. The reference process model P_0.

Fig. 9 shows the reference process model. It describes a real-life insurance claim business as introduced in [38]. For simplicity, the task nodes are represented by single letters.

The main information loss of transforming graphical process models to business process graphs is attributed from the gateways and connectors. It may lead to the same graph branch structure with totally different behavior, i.e., exclusive branch structures and parallel branch structures cannot be distinguished after the transformation. Therefore, we create process variants by modifying the branch structures of the reference process model P_0.

The construction of process variants aims to reflect the behavioral changes based on the branches of the reference model. The basic changes include:

1) deleting branch structures, including exclusive branches with different weights and parallel branches;

2) adding branch structures, including exclusive branches with different weights and parallel branches;

3) changing branches, including changing exclusive branches to parallel branches and changing parallel branches to exclusive branches.

Based on P_0, we constructed thirteen process variants P_1 to P_{13} as shown in Fig. 10.

Then, we generated fourteen process logs for the process models (P_0 to P_{13}). Table 2 shows the basic statistics of the simulated process logs.

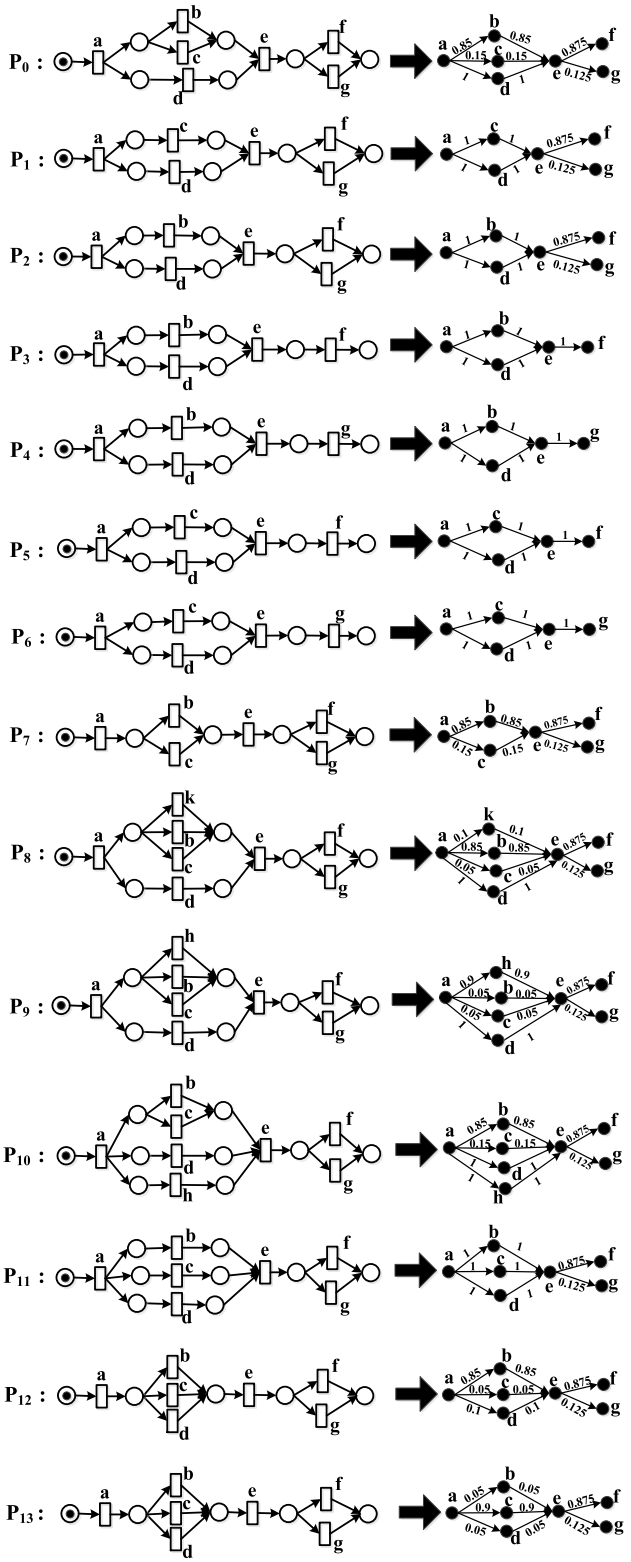


FIGURE 10. Process model $P_0 \sim P_{13}$ and the corresponding *WBPG*.

B. EXPERIMENTAL RESULT

By taking the reference process model, the process variants and the simulated process logs as input, the corresponding *WBPGs* are obtained by Algorithm 1 as shown in Fig. 10.

TABLE 2. Processes simulated logs statistics.

Process \ Log Information	Event Number	Trace Number
P_0	1600	320
P_1	2600	520
P_2	1400	280
P_3	2025	405
P_4	1700	340
P_5	2140	428
P_6	1760	352
P_7	1440	360
P_8	2400	480
P_9	2800	560
P_{10}	2640	440
P_{11}	1440	240
P_{12}	2240	560
P_{13}	640	160

TABLE 3. Experimental group.

Group ID	the processes compared with P_0
G1	P_1, P_2, P_3, P_4, P_5, P_6, P_8, P_9
G2	P_7, P_{10}
G3	P_{11}
G4	P_{12}, P_{13}

1) EXPERIMENTAL PROCESS

The experiment is divided into four groups according to the changes of branching structures as shown in Table 3.

As no existing work considers both process models and process logs to measure similarity, we compare our approaches with traditional process model similarity measures including model structural similarity and model behavioral similarity. Because two of the proposed four approaches are related to the graph edit distance and two are related to the node adjacent relation, we use the graph edit distance [10] approach as the benchmark of traditional structural similarity method. For the behavioral similarity, we choose the PTS [25], the TAR [20] and the ETR [28] methods which are known as the state-of-the-art.

Table 4 summarizes all these approaches where A1-A4 represent the existing traditional approaches and A5-A8 represent our approaches.

2) PARAMETER SETTING

The similarity calculation is executed between the reference process model P_0 (including the process log) and the 13 process variants (including the process logs) for the above-mentioned eight methods. For *GED*, *WGED* and *GED-LBP*, they require a number of parameters, such as *wskipn*, *wskipe, wsube* and *wsubn*, as input. These parameters are

TABLE 4. Process similarity methods comparison.

Approach ID	Algorithm	Measured Target	Reference
A1	*GED*	Model Structural Similarity	[10]
A2	*TAR*	Model Behavioral Similarity	[20]
A3	*PTS*	Model Behavioral Similarity	[25]
A4	*ETR*	Model Behavioral Similarity	[28]
A5	*WGED*	Model Structure and Log behavior comprehensive Similarity	the proposed method
A6	*WNAR*	Model Structure and Log behavior comprehensive Similarity	the proposed method
A7	*NAR-LBP*	Model Structure and Log behavior comprehensive Similarity	the proposed method
A8	*GED-LBP*	Model Structure and Log behavior comprehensive Similarity	the proposed method

determined by running the experiments to find the optimal setting. More specifically, we test different parameter combinations and choose the parameter values that lead to the optimal results. The obtained parameter values for the three approaches are shown in Table 5.

TABLE 5. Parameter values.

Approach ID	Algorithm	wskipn	wskipe	wsube	wsubn
A1	*GED*	0.1	0.1	---	0.4
A5	*WGED*	0.1	0.3	0.3	0.2
A8	*GED-LBP*	0.1	0.1	---	0.4

3) EXPERIMENTAL RESULTS

By taking these parameter values as input, the detail evaluation results are shown in Table 6, based on which we conclude that:

① If the corresponding business process graphs of process models are similar, the similarity maybe identical whatever traditional measures are used. For example, the similarity between P_0 and P_3 (or P_4, P_5 and P_6) is same for *GED*, *TAR*, *ETR* and *PTS*. However, their logs are quite different. Therefore, we can see that existing process model similarity measures cannot illustrate the differences that are attributed from the execution behavior recorded in process logs.

② The information loss when transforming from process models to business process graphs may hide the difference of original processes. For example, P_7 and P_2 are different even if their corresponding *BPGs* are very similar to P_0. However, *GED* and *TAR* are unable to spot the difference and the similarity between P_0 and P_7 (or P_2) is identical. The *PTS* measures the difference from behavioral aspect but ignores the similarity of the model structures.

③ No matter how slightly the differences between process structure and log behavior are, they can be effectively distinguished by our proposed approaches that utilize both process models and process logs.

Then we evaluate the effectiveness of the similarity methods by using the normalized discount cumulative gain (*NDCG*) as introduced in [41]. DCG_n is the score of a ranking order of the first n relevant models, as defined by Equation (5). In our experiment, n equals to 13. Then, *NDCG* is computed by Equations (5)-(6).

$$DCG_n = \begin{cases} r(n), & n = 1; \\ r(1) + \sum_{n=2}^{N} \frac{r(n)}{\log_2 n}, & n > 1; \end{cases} \tag{5}$$

$$NDCG_n = \frac{DCG_n}{IDCG_n} \tag{6}$$

In Equation (5), $r(n)$ is the weight (determined by users) of the *n-th* process model in the ranking order. $IDCG_n$ in Equation (6) refers to the ideal *DCG*, i.e., the maximum value of *DCG*. Essentially, $NDCG_n$ is a normalized *DCG* value. We will use the $NDCG_n$ to evaluate the accuracy of the ranking results.

To obtain the ranking order, we design a user case study. The user case study involves 15 postgraduates who have different areas of expertise, such as service computing, workflow management and machine learning. Each interviewee is asked to rank the order of P_1 to P_{13} models with respect to the reference process model P_0 in terms of similarity. Different interviewees have different ranking results, therefore, we use the following strategy to merge the results.

We assign 13 weights from 0.3 to 0.9 to each ranked model of the 13 process variants, e.g. the model ranked in the first position gets 0.9, the model gets 0.85 if it is ranked in the second position, and so on. Then, the total weight of each variant model is summed up, and the final ranking is determined by their descending order of weights. Then we invited another 10 process experts and 10 process participants to validate the benchmark ranking result. Process experts have a grounded knowledge of the process landscape of a company or its branches, while process participants are specialists for particular processes. Among the 20 invited participants, only 2 participants have different opinions on the ranking result. Therefore, we argue that the benchmark is reliable. The benchmark ranking order and different ranking order by the above eight methods are shown in Table 7.

The accuracy of the ranking order is evaluated and shown in Table 8. Table 8 shows the accuracy of these methods based on Equation (6), where all approaches perform well. It is because all process variants contain the sequence construction that is the simplest construction, and all approaches can deal with them with a high accuracy. Among the eight approaches, *WGED* gets the highest accuracy and *WNAR*

TABLE 6. Experimental results.

Group ID	Group Basis	Process Variants	The similarity with P_0 by different approaches							
			A1	A2	A3	A4	A5	A6	A7	A8
G1	Delete an exclusive branch with different weights	P_1	0.9634	0.75	0.35	0.7	0.8215	0.4925	0.735	0.8417
		P_2	0.9634	0.75	0.35	0.7	0.9615	0.8868	0.85	0.9567
	Delete two exclusive branches with different weights	P_3	0.9338	0.625	0.5875	0.6	0.9389	0.84338	0.7475	0.9019
		P_4	0.9338	0.625	0.5875	0.6	0.8639	0.61948	0.6475	0.8019
		P_5	0.9338	0.625	0.5875	0.6	0.7989	0.4652	0.6525	0.8069
		P_6	0.9338	0.625	0.5875	0.6	0.7239	0.32	0.5625	0.7169
	Add an exclusive branch with different weights	P_8	0.9704	0.8	0.6	0.6	0.9726	0.923	0.865	0.9502
		P_9	0.9704	0.8	0.6	0.77	0.8126	0.47	0.75	0.8352
G2	Delete a parallel branch	P_7	0.9634	0.75	0.244	0.77	0.9081	0.6	0.77	0.8767
	Add a parallel branch	P_{10}	0.9704	0.8	0.2315	0.67	0.937	0.7143	0.79	0.8752
G3	Change an exclusive branch to a parallel branch	P_{11}	1	1	0.1065	0.91	0.9048	0.7143	0.91	0.91
G4	Change a parallel branch to an exclusive branch with different weights	P_{12}	1	1	0.57	0.8	0.8667	0.6	0.9	0.9
		P_{13}	1	1	0.57	0.8	0.7436	0.2308	0.79	0.79

TABLE 7. Standard ranking order and the experimental ranking orders.

Ranking Weight	0.9	0.85	0.8	0.75	0.7	0.65	0.6	0.55	0.5	0.45	0.4	0.35	0.3
Standard Ranking	P_8	P_2	P_3	P_{10}	P_{12}	P_7	P_{11}	P_1	P_9	P_5	P_4	P_6	P_{13}
Experimental ranking order by eight approaches													
Order by A1	P_{11}	P_{12}	P_{13}	P_8	P_9	P_{10}	P_1	P_2	P_7	P_3	P_4	P_5	P_6
Order by A2	P_{11}	P_{12}	P_{13}	P_8	P_9	P_{10}	P_1	P_2	P_7	P_3	P_4	P_5	P_6
Order by A3	P_8	P_9	P_3	P_4	P_5	P_6	P_{12}	P_{13}	P_1	P_2	P_7	P_{10}	P_{11}
Order by A4	P_{11}	P_{12}	P_{13}	P_9	P_7	P_1	P_2	P_{10}	P_3	P_4	P_5	P_6	P_8
Order by A5	P_8	P_2	P_3	P_{10}	P_7	P_{11}	P_{12}	P_4	P_1	P_9	P_5	P_{13}	P_6
Order by A6	P_8	P_2	P_3	P_{10}	P_{11}	P_4	P_7	P_{12}	P_1	P_9	P_5	P_6	P_{13}
Order by A7	P_{11}	P_{12}	P_8	P_2	P_{10}	P_{13}	P_7	P_9	P_3	P_1	P_5	P_4	P_6
Order by A8	P_2	P_8	P_{11}	P_3	P_{12}	P_7	P_{10}	P_1	P_9	P_5	P_4	P_{13}	P_6

has a similar value which is followed by *GED-LBP*, *NAR-LBP*, *PTS*, *GED*, *TAR* and *ETR*. Therefore, we conclude that the proposed four approaches are better than the traditional methods in terms of accuracy.

Then, we increase the change rate of the weight from 0.05 to 0.08, i.e., the model ranked in the first position get 0.98, 0.9 for the second ranked model, and so on. The *NDCG* values are shown in Table 9.

TABLE 8. Accuracy evaluation.

Approach ID	Algorithm	*IDCG*	*DCG*	*NDCG* (accuracy)
A1	*GED*	4.167253	3.705614	88.92%
A2	*TAR*		3.705614	88.92%
A3	*PTS*		3.771915	90.51%
A4	*ETR*		3.628471	87.07%
A5	*WGED*		4.15684	99.75%
A6	*WNAR*		4.14056	99.36%
A7	*NAR-LBP*		3.884279	93.21%
A8	*GED-LBP*		4.119063	98.84%

TABLE 9. Accuracy evaluation after adjustment.

Approach ID	Algorithm	*IDCG*	*DCG*	*NDCG* (accuracy)
A1	*GED*	3.864944	3.126322	80.89%
A2	*TAR*		3.126322	80.89%
A3	*PTS*		3.232404	83.63%
A4	*ETR*		3.002893	77.69%
A5	*WGED*		3.848283	99.57%
A6	*WNAR*		3.822236	98.89%
A7	*NAR-LBP*		3.412187	88.29%
A8	*GED-LBP*		3.78784	98.01%

Based on Table 9, the *NDCG* values have different degrees of decrease. However, the performance of these eight approaches stay unchanged for accuracy. Specially, we can see that: (1) the increase of weight for accuracy has slight impact on *WGED*, *WNAR* and *GED-LBP*; and (2) the impact is much heavier for other approaches, e.g. the accuracy of *GED* decreased from 88.92% to 80.89%.

C. DISCUSSION

This section mainly discusses different application scenarios where the proposed four approaches perform better than traditional approaches.

1) From the accuracy perspective, the similarity measures by binding the process model structure and the log behavior are more consistent with people's cognition than the existing approaches. Among the proposed approaches, *WBPG*-based measures (*WGED* and *WNAR)* achieve the highest accuracy. Therefore, if users aim to pursue high accuracy, the *WGED* measure or the *WNAR* measure may be the best choice.

2) From the computational complexity perspective, the *GED*, *WGED* and *GED-LBP* have a higher computational complexity as the graph matching problem suffers from the NP-hard complexity. Therefore, if users pay more attention to computational complexity, the *WNAR* or the *NAR-LBP* measures may be a better choice than *GED* related approaches.

3) From the flexibility perspective, the *WBPG*-based measures lack flexibility because the ratio of structural information and the log behavioral information in the *WBPG* is fixed. The *WGED* approach is slightly better than the *WNAR* approach because the cost weight is flexible. In comparison, the *LBP*-based measure has higher flexibility with different coefficient settings according to people's preference for the process structure and the process log behavior. Therefore, if users focus on the flexibility, the *NAR-LBP* and the *GED-LBP* measures are more suitable.

VII. EVALUATION OF THE PACKAGE REDUCTION APPROACH

In this paper, we provide two frameworks to measure the process similarity by considering both process models and process logs. In the first framework, heterogeneous processes represented by different graphical notations are uniformly transformed to BPGs. Then, we construct the *WBPG* by adding weights to the edges of *BPGs* according to the logs. Based on the *WBPGs*, we propose the *WBPG*-based approaches, including *WGED* and *WNAR*, to measure the similarity. For the second framework, we borrow the idea of *EMD* to measure the similarity of the logs which can be expressed as a multi-set of sequences. Then, the other two approaches *NAR-LBP* and *GED-LBP* are proposed by computing the model structural similarity and the log similarity separately. Finally, we perform the experiments to evaluate the proposed four approaches and analyze the effectiveness and the application. The experimental results demonstrate that the proposed approaches can better reflect the execution preference which facilitates recommending similar processes with high accuracy.

This work opens the door for the following directions:

1) We plan to incorporate existing (or more advanced) graph matching algorithms to improve the performance of our approaches; and

2) We also plan to explore and evaluate our approaches to real-life process models and process logs with dedicated domain knowledge in the future.

REFERENCES

[1] M. Becker and R. A. Laue, "A comparative survey of business process similarity measures," *Comput. Ind.*, vol. 63, no. 2, pp. 148–167, Feb. 2012.

[2] A. Schoknecht, T. Thaler, P. Fettke, A. Oberweis, and R. Laue, "Similarity of business process modelsŮA state-of-the-art analysis," *ACM Comput. Surv.*, vol. 50, no. 4, p. 52, Nov. 2017.

[3] T. Thaler, A. Schoknecht, P. Fettke, A. Oberweis, and R. Laue, "A comparative analysis of business process model similarity measures," in *Proc. Int. Conf. Bus. Process Manage.*, May 2017, pp. 310–322.

[4] R. Dijkman, M. Dumas, B. V. Dongen, R. Käärik, and J. Mendling, "Similarity of business process models: Metrics and evaluation," *Inf. Syst.*, vol. 36, no. 2, pp. 498–516, Apr. 2011.
[5] U. Cayoglu *et al.*, "Report: The process model matching contest," in *Proc. Int. Conf. Bus. Process Manage.*, May 2014, pp. 442–463.
[6] G. Antunes *et al.*, *The Process Model Matching Contest 2015* (Lecture Notes in Business Information Processing), vol. 171. 2015, pp. 442–463.
[7] R. Akkiraju and A. Ivan, "Discovering business process similarities: An empirical study with SAP best practice business processes," in *Proc. Int. Conf. Service-Oriented Comput.*, Oct. 2010, pp. 515–526.
[8] M. Minor, A. Tartakovski, and R. Bergmann, "Representation and structure-based similarity assessment for agile workflows," in *Proc. Int. Conf. Case-Based Reasoning*, Aug. 2007, pp. 224–238.
[9] X. Gao, B. Xiao, D. Tao, and X. Li, "A survey of graph edit distance," *Pattern Anal. Appl.*, vol. 13, no. 1, pp. 113–129, Feb. 2010.
[10] R. Dijkman, M. Dumas, and L. García-Bañuelos, "Graph matching algorithms for business process model similarity search," *Comput. Sci.*, vol. 57, no. 1, pp. 48–63, 2009.
[11] V. Gacitua-Decar and C. Pahl, "Structural process pattern matching based on graph morphism detection," *Int. J. Softw. Knowl. Eng.*, vol. 27, no. 2, pp. 153–189, May 2017.
[12] J. Li, L. J. Wen, and J. M. Wang, "Process model storage mechanism based on Petri net edit distance," *Comput. Integr. Manuf. Syst.*, vol. 19, no. 8, pp. 1832–1841, 2013.
[13] N. Jia, Y. Huang, Z. H. Dai, X. D. Fu, and X. Y. Liu, "Workflow distance metric based on tree edit distance," *J. Comput. Appl.*, vol. 32, no. 12, pp. 1766–1773, 2012.
[14] X. Fu, K. Yue, P. Zou, F. Wang, and K. Ji, "A process distance metric based on alignment of process structure trees," in *Proc. Asia–Pacific Conf. Web Technol. Appl.*, Apr. 2012, pp. 221–232.
[15] Y. Zhang, J. Liu, and L. Wang, "Product manufacturing process similarity measure based on attributed graph matching," in *Proc. 3rd Int. Conf. Mechatron., Robot. Automat.*, Apr. 2015, pp. 1083–1086.
[16] M. L. Sebu and H. Ciocârlie, "Similarity of business process models in a modular design," in *Proc. IEEE 11th Int. Symp. Appl. Comput. Intell. Inform. (SACI)*, May 2016, pp. 31–36.
[17] B. Dongen, R. Dijkman, and J. Mendling, "Measuring similarity between business process models," in *Proc. Seminal Contrib. Inf. Syst. Eng.*, Jan. 2008, pp. 450–464.
[18] B. Kiepuszewski, A. H. M. T. Hofstede, and C. J. Bussler, "On structured workflow modelling," in *Proc. Int. Conf. Adv. Inf. Syst. Eng.*, Dec. 2000, pp. 431–445.
[19] A. Valmari, "The state explosion problem," in *Proc. Adv. Course Petri Nets*, Jun. 2005, pp. 429–528.
[20] H. Zha, J. Wang, L. Wen, C. Wang, and J. A. Sun, "A workflow net similarity measure based on transition adjacency relations," *Comput. Ind.*, vol. 61, no. 5, pp. 463–471, Jun. 2010.
[21] M. Kunze, M. Weidlich, and M. Weske, "Behavioral similarity—A proper metric," in *Proc. Int. Conf. Bus. Process Manage.*, Aug. 2011, pp. 166–181.
[22] M. Weidlich, J. Mendling, and M. A. Weske, "A foundational approach for managing process variability," in *Proc. Int. Conf. Adv. Inf. Syst. Eng.*, Jun. 2012, pp. 267–282.
[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 2011.
[24] S. Jinfeng, W. Lijie, and W. A. Jianmin, "A similarity measure for process models based on task occurrence relations," *J. Comput. Res. Develop.*, vol. 54, no. 4, pp. 832–843, 2017.
[25] J. Wang, T. He, L. Wen, N. Wu, A. H. M. T. Hofstede, and J. Su, "A behavioral similarity measure between labeled Petri nets based on principal transition sequences," in *Proc. OTM Confederated Int. Conf.*, Oct. 2010, pp. 394–401.
[26] Z. Dong, L. Wen, H. Huang, and J. Wang, "CFS: A behavioral similarity algorithm for process models based on complete firing sequences," in *Proc. OTM Confederated Int. Conf.*, Oct. 2014, pp. 202–219.
[27] Z. X. Wang, L. J. Wen, S. H. Wang, and J. M. Wang, "Similarity measurement for process models based on transition-labeled graph edit distance," *Comput. Integr. Manuf. Syst.*, vol. 22, no. 2, pp. 343–352, 2016.
[28] C. Liu, Q. Zeng, H. Duan, S. Gao, and C. Zhou, "Towards comprehensive support for business process behavior similarity measure," *Trans. Inf. Syst.*, vol. 102, no. 3, pp. 588–597, 2018.
[29] A. K. A. D. Medeiros, W. M. P. V. D. Aalst, and A. J. M. M. Weijters, "Quantifying process equivalence based on observed behavior," *Data Knowl. Eng.*, vol. 64, no. 1, pp. 55–74, Jan. 2008.
[30] W. M. P. V. D. Aalst, "Formalization and verification of event-driven process chains," *Inf. Softw. Technol.*, vol. 41, no. 10, pp. 639–650, Jul. 1999.
[31] R. Eshuis and R. Wieringa, "Tool support for verifying UML activity diagrams," *IEEE Trans. Softw. Eng.*, vol. 30, no. 7, pp. 437–447, Jul. 2004.
[32] M. Weske, *Business Process Management: Concepts, Languages, Architectures*. Heidelberg, Germany: Springer, 2010.
[33] Q. T. Zeng, F. Lu, C. Liu, H. Duan, and C. Zhou, "Modeling and verification for cross-department collaborative business processes using extended Petri nets," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 45, no. 2, pp. 349–362, Feb. 2015.
[34] Q. Zeng, C. Liu, and H. Duan, "Resource conflict detection and removal strategy for nondeterministic emergency response processes using Petri nets," *Enterprise Inf. Syst.*, vol. 10, no. 7, pp. 729–750, Sep. 2015.
[35] 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.*, to be published.
[36] C. Liu, Q. Zeng, H. Duan, M. Zhou, F. Lu, and J. Cheng, "E-Net modeling and analysis of emergency response processes constrained by resources and uncertain durations," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 45, no. 1, pp. 84–96, Jan. 2015.
[37] J. Cheng, C. Liu, M. Zhou, Q. Zeng, and A. Y. Jääski, "Automatic composition of semantic Web services based on fuzzy predicate petri nets," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 2, pp. 680–689, Apr. 2015.
[38] W. M. P. V. D. Aalst, *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Heidelberg, Germany: Springer 2011.
[39] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *Int. J. Comput. Vis.*, vol. 40, no. 2, pp. 99–121, Nov. 2000.
[40] C. Higuera and L. Mico, "A contextual normalised edit distance," in *Proc. IEEE 24th Int. Conf. Data Eng. Workshop*, Apr. 2008, pp. 354–361.
[41] J. Wang, B. Cao, W. An, J. Fan, and J. Yin, "A benchmark dataset for evaluating process similarity search methods," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, Jun. 2017, pp. 914–917.

CHANGHONG ZHOU is currently pursuing the Ph.D. degree in software engineering with the Shandong University of Science and Technology, Qingdao, China.

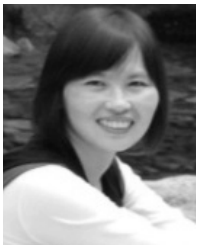
She is currently a Lecturer with the Shandong University of Science and Technology. Her current research interests include process recommendation and process management.

CONG LIU (S'18) received the B.S. and M.S. degrees in computer software and theory from the Shandong University of Science and Technology, Qingdao, China, in 2013 and 2015, respectively. His research interests include business process management, process mining, and Petri nets.

QINGTIAN ZENG received the B.S. and M.S. degrees in computer science from the Shandong University of Science and Technology, Tai'an, China, in 1998 and 2001, respectively, and the Ph.D. degree in computer software and theory from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2005. He is currently a Professor with the Shandong University of Science and Technology, Qingdao, China. His research interests include Petri nets, process mining, and knowledge management.

ZEDONG LIN is currently pursuing the Ph.D. degree in software engineering with the Shandong University of Science and Technology, Qingdao, China.

He is currently an Engineer with the Shandong University of Science and Technology. His current research interests include the Internet public sentiment monitoring and process management.



HUA DUAN received the B.S. and M.S. degrees in applied mathematics from the Shandong University of Science and Technology, Tai'an, China, in 1999 and 2002, and the Ph.D. degree in applied mathematics from Shanghai Jiao Tong University, in 2008. She is currently an Associate Professor with the Shandong University of Science and Technology. Her research interests include Petri nets, business process management, and machine learning.