SimBPG: A Comprehensive Similarity Evaluation Metric for Business Process Graphs

Qinkai Jiang, Jiaxing Wang, Bin Cao(✉), and Jing Fan

Zhejiang University of Technology, Hangzhou 310023, China
{jiangqinkai,wjx,bincao,fanjing}@zjut.edu.cn

Abstract. Measuring the similarity between two business process models holds significant importance across various applications. At present, there are many different similarity calculation methods, such as structural similarity based on the graph edit distance(GED), text similarity based on task node description, and behavioral similarity calculation based on path matching. However, existing similarity computation methods cannot produce reliable results since: (1) To apply GED, business process graphs will be simplified to homogeneous graph where the heterogeneity as well as the routing semantics of the business process is removed. (2) To derive comprehensive similarity evaluation, linear weighted sum of different similarity metrics is a common way, but the final result strongly depends on the weighting coefficients that are empirically assigned. In this paper, we fuse multidimensional metrics to compensate for the sole reliance on structural similarity based on GED. To address the limitations of comprehensive evaluation, we propose a novel multidimensional process similarity evaluation method based on the entropy weight method and the Technique for Order Preference by Similarity to an Ideal Solution (TOPSIS) method. We also design a experimental method to verify the effectiveness of our method, leveraging an open source dataset. The experiment shows that our method can better represent the similarity of business process graphs than other methods.

Keywords: business process graphs · evaluation metric · similarity calculation · heterogeneous graphs · KM algorithm

1 Introduction

Business process management [1] technology has been applied in many enterprises, and more and more enterprises have built their own business process library. With the development of business, the scale of the business process library is also getting larger and larger, how to calculate the business process similarity has become an important link in the management of these business process libraries [2–4]. In general, business processes are often modeled as business process graphs, which are used to describe the business relationship between various units and personnel in the management system, the sequence of operations and the flow of management information [5,6].

H. Gao et al. (Eds.): CollaborateCom 2023, LNICST 562, pp. 437–457, 2024.
https://doi.org/10.1007/978-3-031-54528-3_24

In the field of business process similarity calculation, most of the previous work [7–9] have proposed to measure business process graphs similarity according to different business process requirements, such as structural similarity based on the graph edit distance(GED) [10], text similarity based on task node description [11,12], and behavioral similarity calculation based on path matching [13,14]. However, this single-dimension similarity measure cannot reflect the overall similarity of the business process graph. As far as we know, with the rise of graph neural networks (GNN) [15], more and more methods use graph embedding and graph matching methods [16,17] to train a deep learning model to calculate the similarity score between graphs. But they still use the calculation results of GED as the Ground Truth of the similarity between the two graphs, they do not essentially consider whether the true value of the model fitting is reliable, which will lead to the inaccuracy of the trained model. Actually, GED need to simplify the original business process graph into an isomorphic graph, which does not distinguish between the types of task nodes and the different execution sequences between them. This simplification ignores the important characteristic that the business process graph is essentially a heterogeneous graph [18,19], which makes the routing semantics of the original business process graph lost, and cannot accurately describe the control flow semantics of the original process graph, resulting in low reliability of the similarity calculation results. Therefore, adopting a multi-dimensional comprehensive evaluation method is more helpful to obtain the similarity between business process graphs.

Furthermore, there are some works [20,21] that consider process similarity information from multiple dimensions comprehensively, the existing multi-dimension evaluation fusion methods employ a linear weighted sum of different similarity metrics, which depends on the weighting coefficients that are empirically assigned. Although these methods combine information from multiple dimensions, they lack the adaptability to capture complex relationships and nuances in the data. Therefore, the reliability and robustness of the overall similarity assessment may be compromised. To address these challenges, there is an increasing urgency to develop a more complex and adaptable approach that holistically integrates multidimensional process similarity information while alleviating the reliance on empirically defined weights.

In this paper, we disassemble the heterogeneous information of the business process graph, separate the similarity metrics of different dimensions, and perform effective fusion to obtain the similarity score. Specifically, our method consists of the following two parts:(1) *Adaptive Weights Assignment.* We evaluate the similarity of business process graphs from multiple dimensions, and calculate the similarity scores of different dimensions, such as structural similarity based on GED method and behavioral similarity based on path matching [13]. Then we use the entropy weight method [22] to objectively assign weights to metrics of multiple dimensions to remove the influence of human subjectivity. (2) *Comprehensive Evaluation with TOPSIS.* Based on the similarity metrics of different dimensions obtained in the previous phase. Then use the ideal solution similarity ranking technique (TOPSIS) method [23] to fuse the similarity of multiple dimensions to get the final similarity score.

In the experimental part, since it is impossible to define the real similarity scores of two graphs, there is no Ground Truth process similarity calculation dataset. So we designed an experiment to verify the effectiveness of our method. Our experimental data comes from real business process models collected by IBM [24], which involve different domains and different versions of models. Based on the characteristics of this dataset, we particularly designed a set of experiments to verify the proposed comprehensive similarity measuring method. Specifically, our experimental design involves following three assumptions: (1) *The two business processes in different domains are likely to be dissimilar.* For example, the insurance claims process is different from the bank loan process. (2) *The similarity between different business processes is smaller than that of different versions of the same business process.* For example, different versions of a bank loan process will contain some common sequence of execution steps. (3) *The closer the version number, the smaller the change, and the greater the difference between the version numbers, the greater the change.* Then we conduct three sets of experiments for validation, confirming the reliability of our assumptions.

We summarize our contributions as follows:

– We comprehensively evaluate the similarity of different dimensions of business process graphs using the entropy weight method and the Technique for Order Preference by Similarity to an Ideal Solution (TOPSIS).
– In order to verify the proposed method, we propose three assumptions based on real datasets, and design cross-domain experimental to verify the effectiveness of evaluation metrics.
– Proven by extensive experiments, our method can better characterize the similarity measurement problem of business process graphs than traditional GED or behavioral similarity only.

The remainder of this paper is organized as follows. Section 2 gives the related work. Section 3 presents some preliminaries for our work. Section 4 provides a detailed description of our algorithm. The experimental and concludes are given in Sect. 5 and Sect. 6 respectively.

2 Related Work

As far as we know, no in previous studies has proposed a standard for measuring the similarity of heterogeneous graphs. A lot of work has only focused on representation learning for heterogeneous graphs [18,19,25,26]. However, in the field of business process graph similarity measurement, a lot of work focuses on the similarity calculation of business process model, which mainly divide into three aspects: (1) Text similarity considering task semantics information. Pamungkas E W et al. [11] attempted to use word sense disambiguation to improve the accuracy of business process similarity calculations. Akkiraju et al. [12] measured similarity of business process models only based on the number of equally labeled activities. (2) Structural similarity based on business process model. The most popular approach is to use Petri nets [27] to model and calculate the structural similarity between them. Li J et al. [8] used the greedy algorithm to calculate the GED between different process models based on the Petri net. Sebu et al. [9] compared the graphs considering the composition of the subgraphs and extracted the business process similarity factor. (3) Behavior similarity based on process mining. Cao B et al. [13] introduced the idea of fundamental path testing from the field of software testing and proposed an effective method to detect differences in workflow behavior. Wang Z X et al. [14] expressed the behavior of the process model by defining a transitional marker graph, refine the graph editing operation and the calculation method of the GED according to the behavior characteristics. However, they are all only consider one aspect of business process similarity, therefore, their method can only be calculated for the similarity of a particular business process requirement, and is not universal and accurate.

A small number of methods have tried multi-dimension fusion. Cao B et al. [7] proposed a query method based on the Hungarian algorithm, which defines the contextual similarity of a pair of place nodes from different process models by considering common paths and common transitions, and maps to the classic assignment problem that the Hungarian algorithm can effectively solve. They integrate structural similarity and behavioral similarity, but their methods only consider the behavioral similarity of local optimal matching, not the global behavior of the business process, which has an impact on evaluating the similarity of the entire business process. Aisyah et al. [20] and Zhou et al. [21] conducted a comprehensive evaluation of the similarity of business process graphs. They considered the weight differences between different branches and performed simple addition or linear changes to obtain comprehensive considerations of similarity in different dimensions. However, this method relying on empirical weights cannot reflect the similarity in the real multi-dimension space.

3 Preliminaries

This section we introduce the concepts of *Business Process Graph*, *Basic Path* and *Edit Distance*, which help to understand our method.

3.1 Business Process Graph

Business process model is often used to help identify, describe and decompose business processes. It can be modeled in many ways, including Event-driven Process Chains (EPC), UML Activity Diagrams, Business Process Modeling Notation (BPMN) and Petri nets [5,28]. In this paper, we only consider the similarity calculation between the two business process, trying to find a general graph similarity evaluation metric, the previous business process model representation method is too complex and the detail representation is too specific, which is not suitable for the similarity calculation of the business processes, so we define a new representation similar to Petri nets [29], but simpler and clearer than Petri nets.

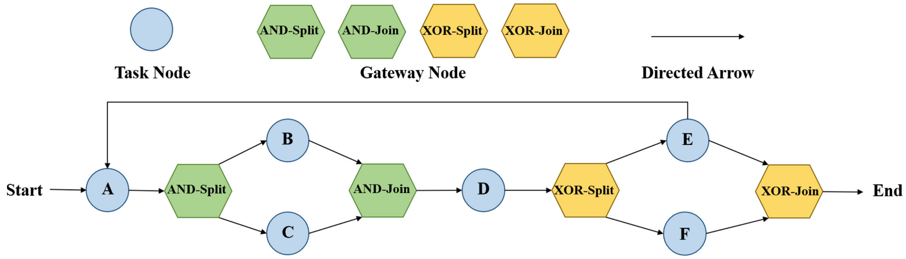


Fig. 1. Example of a business process graph.

Definition 1: a business process graph (BPG) is a 3-tuple *BPG = (T,G,E)*, where (1) T is a set of the task nodes, (2) G is a set of gateway nodes, (3) E is a set of directed arrows.

The gateway node has four types: AND-Split, AND-Join, XOR-Split and XOR-Join. When multiple tasks need to be executed in parallel, AND-Split and AND-Join need to be used. When mutually exclusive selection needs to be made, XOR-Split and XOR-Join need to be used. And the direction of the arrow indicates the direction of sequential execution. A example is shown in Fig. 1.

3.2 Basic Path

In the field of software testing [30], the basic path is based on the program control flow graph, by analyzing the loop complexity of the control structure, deriving the set of basic executable paths, and then designing the corresponding test cases [31] [32]. Inspired by basic path testing, we can extract independent paths from business process graphs and treat them as behavioral information of the graph. We calculate the cyclomatic complexity [33] to measure the complexity of the entire business process, and use this measure as the number of basic paths. The calculation method is as follows:

$$V(G) = |E| - |V| + 2 \tag{1}$$

where *V(G)* denotes the cyclomatic complexity, $|E|$ denotes the number of edges, the $|V|$ denotes the number of nodes.

Since there are many loops in the business process graph, we have specially processed these paths, that is, adding the $'loop'$ string and the next task node after the loop starts. For example, for the business process graph in Fig. 1, we can find two basic paths, which are $\{A, B, C, D, E, 'loop', A\}$ and $\{A, B, C, D, F\}$. These two paths represent the process of business process execution. Here, for the sake of convenience, we put the B and C tasks of parallel operations into the basic path in sequence, but this does not have a great impact on the execution process of the entire business process graph, because both tasks B and C have completed their execution process before task D.

3.3 Edit Distance

The edit distance is used to calculate the distance between two isomorphic graphs or two linear basic paths, it is defined as the minimum cost of transforming a graph (or basic path) into another graph (or basic path) through various transformations. The edit distance is divided into graph edit distance and path edit distance.

Graph Edit Distance(GED) [10] has been widely used in many applications, such as graph similarity search, graph classification, handwriting recognition, image indexing, etc. For two graphs G_1 and G_2, the graph edit distance GED(G_1,G_2) refers to the minimum number of operations required to complete the mutual conversion between them through the insert, delete, and substitute operations of nodes and edges. Figure 2 shows an example of GED between two simple graphs. The GED between the graph to the left and the graph to the right is 3, as the transformation needs 3 edit operations: (1) delete an edge, (2) insert an edge, and (3) relabel a node.

Path Edit Distance(PED). Similar to GED, for two basic paths P_1 and P_2, we define PED(P_1,P_2) as the minimum number of operations for converting P_1 to P_2, and the operation is only for inserting, deleting and substituting nodes. For example, suppose $P_1 = \{A, B, C, D, E, M\}$, $P_2 = \{A, G, B, C, D, F\}$, the basic path P_1 to P_2 require 3 edit operations: (1) insert node G, path converted to $\{A, G, B, C, D, E, M\}$, (2) delete node E, path converted to $\{A, G, B, C, D, M\}$, (3) replace node M with F, path converted to $\{A, G, B, C, D, F\}$.

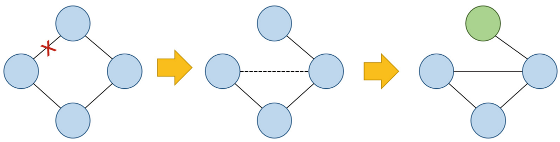


Fig. 2. Example of Graph Edit Distance.

4 Method

In this section, we introduce our proposed method SimBPG in detail. The input is any two business process graphs, while the output is the similarity score between them. Since there are multiple node types in the business process graph, we can regard it as a special heterogeneous graph. Our method is divided into two steps: *Adaptive Weights Assignment* and *Comprehensive Evaluation with TOPSIS*. The overall framework is shown in Fig. 3. First, calculate the similarity of different dimensions of two business process graphs by different methods, such as structural similarity based on GED [10] and behavioral similarity based on global semantic routing [13]. Then use the method based on entropy weight method [22] and the Technique for Order Preference by Similarity to an Ideal Solution (TOPSIS) [23] method to effectively fuse the features of each dimension.

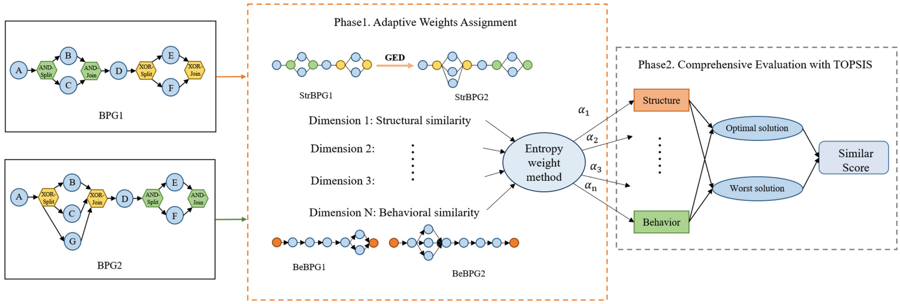


Fig. 3. An overview of SimBPG.

4.1 Phase1. Adaptive Weights Assignment

Multidimensional Similarity Calculation. The evaluation of the similarity of business process graphs can be carried out from multiple perspectives, which helps to provide valuable information for the optimization, improvement and comparison of business processes. For example, structural similarity helps to understand the basic framework and logic of business processes, and behavioral similarity is very important for identifying differences between different versions of graphs. We will detail the calculation methods of structural similarity and behavioral similarity we used in our experiments in the section, so that readers can have a more vivid understanding of our method.

Structural Similarity Calculation. In previous work, the structural similarity of graphs is usually calculated using graph edit distance (GED) [10]. The GED mainly measures the matching degree between graphs by measuring the dissimilarity between graphs, and the dissimilarity is measured by the distance value. The larger the distance value is, the greater the dissimilarity is, and the lower the matching degree is. However, they are suitable for isomorphic graphs and not for heterogeneous graphs. Therefore, we regard all nodes in the graph as nodes of the same type, and only consider the label information of nodes. Then we calculate the structural similarity of two graphs by the following formula:

$$Structure(G_1, G_2) = \frac{1}{1 + GED(G_1, G_2)} \tag{2}$$

where $GED(G_1, G_2)$ represents the graph edit distance between G_1 and G_2.

For BPG1 and BPG2 in Fig. 3, we convert them into StrBPG1 and StrBPG2 respectively, calculate the GED between them as 7, that is, adding 2 edges and 1 node, relabeling 4 nodes and substitution formula (2) can calculate the structural similarity between them as 0.125.

Behavioral Similarity Calculation. Inspired by workflow difference detection based on basis paths method [13] [30], we think that the execution path of the business process graph can cover all node and edge information, so it can be used to represent global behavior information. We find out basic paths in the two graphs, and use path edit distance to represent the difference distance of two paths. However, there are different numbers of paths in each graph, how to calculate the composite similarity score between multiple paths is a challenge. To solve this problem, we use the KM algorithm [34] to search for the best mapping of the two path set paths, and finally calculate the behavioral similarity score of the two graphs. The specific method is divided into the following three steps.

Step1. Find the basic path: There are the following two steps for finding the basic path: (1) Add start and end nodes. In order to find the path, we must find the start and end nodes, so we define a start node uniformly and look for all paths from this node until the end node. (2) Delete the gateway node, and add the nodes after the AND gateway in order. For the path, the gateway node is not important, we need to focus on the execution path of the task node. At the same time, in order to improve the computational efficiency, we arrange the nodes after the AND gateway in order, which may shorten the path edit distance and thus reduce the overhead.

According to the above calculation method, we can use formula (1) to get *V(G)* independent paths, which represent the behavior characteristic information of each business process graph. For example, BeBPG1 and BeBPG2 in Fig. 3 are converted from BPG1 and BPG2. for BeBPG1, we can get two paths: $\{A, B, C, D, E\}$ and $\{A, B, C, D, F\}$. For BeBPG2, we can get three paths: $\{A, B, D, E, F\}$, $\{A, C, D, E, F\}$ and $\{A, G, D, E, F\}$.

Step2. Construct Behavioral Distance Matrix: We get two different sets of basic paths from two different graphs, and when calculating the similarity, we establish a connection between two different basic paths for all possible combinations. So, we construct a behavioral distance matrix to measure the similarity of independent paths in different business process graphs, which is critical for us to find the maximum matching of the basic paths of two graphs later. Assume that the basic path sets of two graphs are P, Q. We compute the behavioral distance matrix W as follows:

$$W[i][j] = \frac{1}{1 + PED(P_i, Q_j)} \tag{3}$$

where $P_i \in P$ and $P_j \in Q$. $PED(P_i, Q_j)$ represents the path edit distance between the i th path in P and the j th path in Q. $W[i][j]$ represents the behavioral similarity between the i th path in P and the j th path in Q.

Now, we can compute the behavioral distance matrix for BeBPG1 and BeBPG2 as show in Table 1. Since BeBPG1 has 2 paths and BeBPG2 has 3 paths, a 2×3 behavioral distance matrix is constructed with a total of 6 distance values.

Table 1. The behavioral distance matrix for BeBPG1 and BeBPG2

	{A, B, D, E, F}	{A, C, D, E, F}	{A, G, D, E, F}
{A, B, C, D, E}	0.33	0.33	0.2
{A, B, C, D, F}	0.33	0.33	0.2

Step3. Optimal Matching Based on KM Algorithm: By calculating the behavioral similarity matrix, we get the behavioral difference score between different independent paths in the two business process graphs, which is equivalent to establishing all possible connections of the two sets. However, there are many possibilities for their combination, and we want to find the best mapping with the largest behavioral difference score to measure the behavioral similarity between two business process graphs. This is actually the optimal matching problem of a weighted bipartite graph [35], which we solve through the KM algorithm [34].

We optimally match the basic path difference scores of the two graphs. Suppose the path set P has i nodes, and Q has j nodes. We match the nodes in P with the nodes in Q, so that the sum of $W[i][j]$ in the matching is the largest, which is defined as $MaxWeight$ and obtained by the km algorithm [34]. Next, we calculate the behavioral similarity of the two graphs as:

$$Behavior(G_1, G_2) = \frac{MaxWeight}{min(i, j)} \tag{4}$$

where $MaxWeight$ is the weight sum of the best match.

In the example, the $\{A, B, C, D, E\}$ and $\{A, B, C, D, F\}$ of BeBPG1 are matched with $\{A, B, D, E, F\}$ and $\{A, C, D, E, F\}$ in BeBPG2 respectively, where $MaxWeight = 0.33$ and the behavioral similarity between them is 0.33.

Entropy Weight Method. Now, we get similarity scores in multiple dimensions, namely structural similarity $Structure(G_1, G_2)$ and behavioral similarity $Behavior(G_1, G_2)$. It is inaccurate to simply add two values together to get the final similarity score. Structural similarity focuses on the static structure of business processes, such as the way nodes and edges are connected, while behavioral similarity focuses on the dynamic execution trajectory of business processes, such as the execution order of different nodes. Simply adding them up will make the structure and behavior information be treated equally in the similarity calculation, which may lead to information redundancy. Also, simple addition may result in loss of information. Because the structural and behavioral similarities may take values on different magnitudes, simple summing will make the similarity with high values have a greater impact on the final result, thereby ignoring the similarities with low values. In order to avoid information redundancy and information loss, the method of fusing structural and behavioral similarities needs to consider their weight and importance.

Due to the complexity of business processes, the business differences of different datasets may lead to different contributions of similarity metrics in different dimensions to the final similarity score. Therefore, we objectively weight the similarity indicators of different dimensions based on specific datasets. We draw on the idea of information entropy [22], which calculates the information entropy of the indicator, and determines the weight of the indicator according to the impact of the relative change degree of the indicator on the overall system. Indicators with a large degree have a greater weight. The specific implementation scheme is as follows.

Suppose we have n groups to calculate structural similarity and behavioral similarity, so we get the following matrix X of $n \times 2$, where x_{ij} represents the value of the j th evaluation index of the i th sample. We calculate the final weight by the following formula.

$$p_{ij} = \frac{x_{ij}}{\sum_{i=1}^{n} x_{ij}} \tag{5}$$

$$e_j = -\frac{1}{\ln n}\sum_{i=1}^{n} p_{ij} \ln p_{ij}, e_j \in [0,1] \tag{6}$$

$$d_j = 1 - e_j \tag{7}$$

$$\alpha = \frac{d_1}{d_1 + d_2} \tag{8}$$

$$\beta = \frac{d_1}{d_1 + d_2} \tag{9}$$

where p represents the probability matrix, e represents the information entropy of each metric, and d represents the information utility value. We get the weight of structural similarity as α and the weight of behavioral similarity as β.

4.2 Phase2. Comprehensive Evaluation with TOPSIS

Next, we need to comprehensively consider multiple dimensions and take into account the weights and values of each dimension to provide a comprehensive evaluation score for the current sample, which can reflect the differences between different pairs of business process graphs. In fact, the similarity of business process graphs is a relative concept and needs to be referenced based on a maximum and minimum value. Therefore, how to quantify the relative distance between each sample and the maximum and minimum values to provide a relatively objective score is a challenge. We use the Technique for Order Preference by Similarity to an Ideal Solution (TOPSIS) [23] method for multi-dimensional fusion, which sorts by comparing the similarity between the sample and the optimal solution and the worst solution, which can comprehensively consider the influence of each dimension, and convert the evaluation criteria of multiple dimensions into a comprehensive sorting result. The value range of the similarity of the comprehensive evaluation is [0,1], where 0 and 1 indicate that the two graphs are completely different and exactly the same, respectively. In general, the optimal solution of the same dimension of two business flow charts is 1, and the worst solution is 0. So, we calculate the distances of the obtained $Structure(G_1, G_2)$ and $Behavior(G_1, G_2)$ from the optimal solution and the worst solution respectively:

$$d^+(G_1, G_2) = \sqrt{(1 - \alpha Structure(G_1, G_2))^2 + (1 - \beta Behavior(G_1, G_2))^2} \tag{10}$$

$$d^-(G_1, G_2) = \sqrt{\alpha Structure(G_1, G_2)^2 + \beta Behavior(G_1, G_2)^2} \tag{11}$$

where d^+ represents the distance between the two dimensions and the optimal solution, d^- represents the distance between the two dimensions and the worst solution. α and β represent the weights of each dimension, and their values are between [0,1], and $\alpha + \beta = 1$.

Finally, we calculate the score based on the optimal solution and the worst solution:

$$SimilarScore(G_1, G_2) = \frac{2d^-(G_1, G_2)}{d^+(G_1, G_2) + d^-(G_1, G_2)} \tag{12}$$

In the example, when $\alpha = 0.5$ and $\beta = 0.5$, then $d^+ = 1.26$, $d^- = 0.25$, the final similarity score is 0.33. The proposed method calculation process is summarized as Algorithm 1.

Algorithm 1: SimBPG calculation method

Input: A pair of business process graphs G_1, G_2
Output: The similarity between two business process graphs: $SimilarScore$

1 $Structure(G_1, G_2) \leftarrow$ Calculate structural similarity;
2 $P_1 \leftarrow$ get the basic path from G_1;
3 $P_2 \leftarrow$ get the basic path from G_2;
4 $W \leftarrow$ initialize the behavior distance matrix;
5 **for** $i \leftarrow 0$ **to** $P.length()$ **do**
6   **for** $j \leftarrow 0$ **to** $Q.length()$ **do**
7         $PED(P_i, Q_j) \leftarrow$ Calculate the path distance;
8         $W[i][j] \leftarrow \frac{1}{1+PED(P_i, Q_j)}$;
9   **end**
10 **end**
11 $MaxWeight \leftarrow KM(P, Q, W)$;
12 $Behavior(G_1, G_2) \leftarrow$ Calculate behavioral Similarity;
13 $\alpha, \beta \leftarrow$ Entropy weight method;
14 $d^+ \leftarrow \sqrt{(1 - \alpha Structure(G_1, G_2))^2 + (1 - \beta Behavior(G_1, G_2))^2}$;
15 $d^- \leftarrow \sqrt{\alpha Structure(G_1, G_2)^2 + \beta Behavior(G_1, G_2)^2}$;
16 $SimilarScore(G_1, G_2) \leftarrow \frac{2d^-(G_1, G_2)}{d^+(G_1, G_2) + d^-(G_1, G_2)}$;
17 return $SimilarScore(G_1, G_2)$

Algorithm 1.shows the flow of our method. The input is two business process graphs G_1, G_2, and the output is Comprehensive similarity score $SimilarScore$. First, calculate the structural similarity of two business process graphs (line 1). Second, obtain the basic paths of the two graphs and construct a behavioral distance matrix (lines 2-10), using the KM algorithm for optimal path matching (line 11) and calculate the behavioral similarity of two business process graphs (line 12). Third, calculate the weight of each dimension based on the entropy weight method (line 13), and calculate the final similarity score based on the TOPSIS method (lines 14-16).

5 Experiment

In this section, we design experiments to prove the effectiveness of the proposed method. There are two major challenges with this experiment. (1) We do not know the true similarity scores of the two business process graphs, so even if we use our method to calculate the similarity between them, we still cannot prove whether the results of the calculation are accurate. (2) Since we do not know the accuracy of the calculation results, we can not compare them with other methods to prove whether the proposed method is effective. In fact, there is no good solution to the above challenges. We can manually judge the similarity between two graphs, but the specific similarity score between them cannot be quantitatively determined, so we propose a cross-domain and multi-dimension experimental design scheme to qualitatively analyze whether our method is effective. Based on the dataset publicly available from IBM Corporation [24], one of the most commonly used datasets in the field of business process modeling, which originates from business processes in different domains. We put forward three assumptions and proved our conjecture through experiments. We also discuss in detail the reasonableness of parameter settings in the method and the influence of the number of nodes and the number of basic paths on the results. All experiments ran on a Windows 11 machine that used AMD Ryzen 9 5900HX with Radeon Graphics, 3.30 GHz, and 16 GB of RAM.

5.1 Dataset

The dataset publicly available from IBM Corporation [24] is a real-world dataset which has more than 3,000 business process models. Due to the incompleteness of some models in the dataset, we extracted more complete 845 different models from the dataset in the fields of insurance, banking, customer relations, and construction and automotive supply chains. We transform these models into the business process graphs we need, and anonymize the data in these models, because we do not consider the textual semantics information. There are 5 libraries in this dataset, including A, B1, B2, B3, and C, and each library represents a model in a different domain. Among them, libraries B1, B2, and B3 have partial overlap, they represent a series of models created in the same field in two years. One of the libraries is changed to the next by adding more process models and further refining all models, B3 is the latest library. We count the number of business process models in the dataset used, including 216 models in library A, 103 models in library B1, 136 models in B2, 255 models in library B3, and 135 models in library C. Table 2 calculates the average number of task nodes and the average number of basic paths in each library.

Table 2. Experimental data

	A	B1	B2	B3	C
Graphs number	216	103	136	255	135
Node-total avg	28.64	19.04	19.36	20.34	12.3
Path-total avg.	2.02	2.28	1.92	2.03	5.21

5.2 Design of Experiments

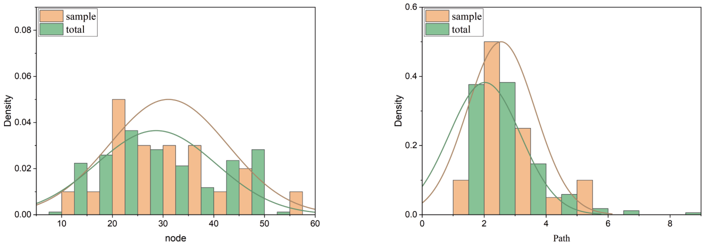
We make three assumptions as follows:

Assumption 1. Assume that business process graphs within the same domain are more similar than between different domains, because the business process graphs in the same domain have similar application scenarios, the semantics information of their nodes and the process execution process have a strong correlation. We define domain similarity as the average similarity score of all business process graphs between domains, and use this value to represent the degree of similarity between domains.

Assumption 2. In the dataset, the same business process graph will be updated into different versions over time, which we name as initial version, intermediate version and final version. Assume that for the same business process graph, it is more similar to other versions than it is to other business process graphs. This is because the same business process graph is created and updated for a specific task requirement, and the structure and behavior information of the business process is only partially adjusted, obviously there is a greater similarity.

Assumption 3. Based on the second assumption, we can assume that the initial and final versions of the same business process graph are less similar than the intermediate versions and other versions. Because the current version of the business process graph is updated and improved based on the previous version of the business process graph, there is a stronger relationship between adjacent versions.

Domain Similarity Experiment. Based on the Assumption 1, we will calculate the average business process graph similarity between the same domain and different domains. Since GED calculation takes a lot of time [7], especially for graphs with a large number of nodes, it is obviously unnecessary to calculate all graphs in the dataset. So we randomly select some graphs for testing. We randomly repeat multiple times to select μ models from different libraries, and construct μ^2 pairs of graphs between the same library and different libraries respectively, and take the average value as the final experimental result. As shown in Fig. 4. Sample represents the sampling data, and total represents the total data. Figure 4(a) and Fig. 4(b) respectively represent the frequency distribution histograms of the number of nodes and the number of paths in library A. We can see that the curves of the two are roughly the same, so we suppose that the sampled data can objectively display the information of all data.



(a) Node number distribution (b) Path number distribution

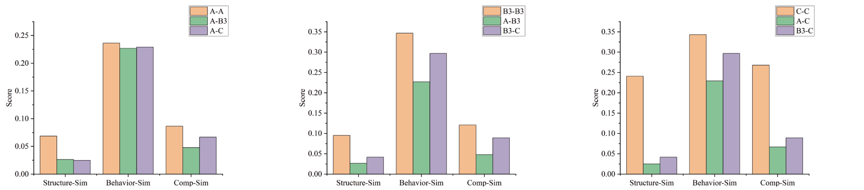
Fig. 4. When $\mu = 20$, node and path distribution of sampled data and overall data in library A

Specifically, we calculate the structural similarity (*Structure-Sim*), behavioral similarity (*Behavior-Sim*) and comprehensive evaluation similarity (*Comp-Sim*) between library A and library A, library A and library B3, library A and library C, library B3 and library B3, library B3 and library C, library C and library C respectively, and the results are shown in Table 3. Here, we designed three sets of experiments, which are $\mu = 20$, $\mu = 30$ and $\mu = 40$, for the business process graph of a certain library, its similarity with the graph in its own library is often higher than that between different libraries. For example, when $\mu = 20$, the average similar score in library A is 0.0865, which is higher than the average similar score with B3 of 0.0479 and the average similar score with C of 0.0667. All the results show a higher degree of similarity between the same libraries, which is in line with our perception that business process graphs in the same domain will be more similar.

Since the structure and behavior information of business process graphs in the same domain are more similar, the calculation using GED can also prove the approximate result of the first assumptions. However, our method has obvious advantages in different domain similarity measures. Since there is no positive relationship between structural and behavioral similarities across domains, the overall difference between the two graphs cannot be accurately measured if only GED is used. For example, when $\mu = 20$, in library A, their average *Structure-Sim* with library B3 graphs is higher than that of library C, and if only measured by GED, the graph of library A and library B3 is more similar. But in fact, whether in terms of the number of task nodes or the complexity of the graph, we tend to think that library A is more similar to library C. So when we consider behavioral similarity, plus *Behavior-Sim*, we find that library A has a higher average similarity score to library C.

Table 3. Domain similarity calculation results between different libraries

	metric	A-A	A-B3	A-C	B3-B3	B3-C	C-C
$\mu = 20$	Structure-Sim	0.0686	0.0265	0.0249	0.0952	0.0416	0.2407
	Behavior-Sim	0.2363	0.2269	0.2292	0.3464	0.2969	0.3431
	Comp-Sim	**0.0865**	**0.0479**	**0.0667**	**0.1208**	**0.0893**	**0.2680**
$\mu = 30$	Structure-Sim	0.0527	0.0250	0.0202	0.0616	0.0291	0.2431
	Behavior-Sim	0.2332	0.2078	0.2278	0.2375	0.2400	0.3421
	Comp-Sim	**0.0686**	**0.0394**	**0.0408**	**0.0815**	**0.0790**	**0.2649**
$\mu = 40$	Structure-Sim	0.0435	0.0352	0.0228	0.1032	0.0393	0.1834
	Behavior-Sim	0.2307	0.2338	0.2248	0.2918	0.2694	0.3029
	Comp-Sim	**0.0571**	**0.0480**	**0.0507**	**0.1240**	**0.0975**	**0.2074**



(a) Domain similarity between A and others (b) Domain similarity between B3 and others (c) Domain similarity between C and others

Fig. 5. When $\mu = 20$, the similarity between a library and other libraries compares the results

Different Graph Similarity Experiments. Based on the Assumption 2, we need to find different versions of the same business process graph. However, due to the anonymization of the task nodes of the dataset, it is impossible to find an updated iterative version of each graph, so we define an error value to determine whether both graphs are the same graph by the following formula.

$$Error(G_1, G_2) = \frac{|(node(G_1) - node(G_2)|}{max(node(G_1), node(G_2))} \tag{13}$$

where $node(G_1)$ and $node(G_2)$ represent the number of nodes in business process graph G_1 and G_2.

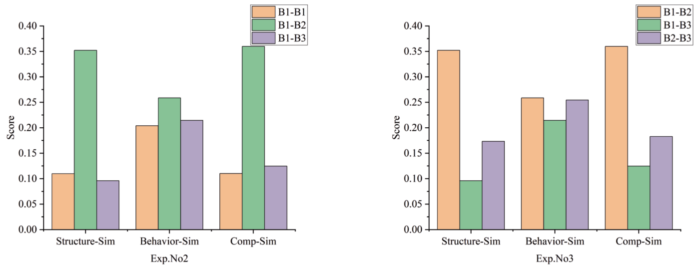
We assume that the error of the two business process graphs within the range of σ represents the same business process graphs. Then, we can find different business process graphs in the same library and the different versions of the same business process graph in different libraries. Due to the similar structure of the experiments in different groups, we randomly select data $\mu = 20$ for experimentation, we first randomly select 20 business process graphs in library B1, and then find one different version in library B2 and B3. We calculate the average similarity of 20 different graphs in library B1 and the average similarity of these 20 graphs between different versions in library B2 and library B3, and the results are shown in Table 4. We set three error values, $\sigma = 0.05$, $\sigma = 0.10$ and $\sigma = 0.15$, and we find that no matter how we set the error values, the results are in line with our assumptions.

However, our assumptions cannot be obtained using only GED. As shown in Fig. 6(a), When $\mu = 20$, $\sigma = 0.05$, the *Structure-Sim* between different graphs in library B1 is higher than the *Structure-Sim* between these graphs and different versions in library B3, and the opposite is true for the final similarity *Comp-Sim*. This is because in the same domain, the application scenarios of the process business graphs are roughly the same, and it is likely to have similar structural information, but the specific execution process in the graph is different. If only GED is used for calculation, then the The similarity of is likely to be greater than the similarity between different versions. However, due to the difference in behavioral information, the final similarity between the two graphs weakens the structural similarity information, and will not be completely dominated by structural similarity, so our method is more reasonable.

Table 4. Similarity calculation results between different versions

	metric	B1-B1	B1-B2	B1-B3	B2-B2	B2-B3	B3-B3
$\sigma = 0.05$	Structure-Sim	0.1097	0.3519	0.0959	0.0702	0.1731	0.0719
	Behavior-Sim	0.2041	0.2585	0.2145	0.2242	0.2544	0.2072
	Comp-Sim	**0.1100**	**0.3597**	**0.1247**	**0.0935**	**0.1827**	**0.0934**
$\sigma = 0.10$	Structure-Sim	0.1097	0.3212	0.1023	0.0634	0.1649	0.0933
	Behavior-Sim	0.2041	0.2364	0.2174	0.2456	0.2456	0.2086
	Comp-Sim	**0.1100**	**0.3263**	**0.1283**	**0.0898**	**0.1983**	**0.0954**
$\sigma = 0.15$	Structure-Sim	0.1097	0.3084	0.0824	0.0547	0.1573	0.0806
	Behavior-Sim	0.2041	0.2156	0.2032	0.2318	0.2358	0.2013
	Comp-Sim	**0.1100**	**0.2983**	**0.1269**	**0.0654**	**0.1828**	**0.0864**

Version Similarity Experiment. Based on the Assumption 3, we believe that the longer the time span of different versions of the same business process graph, the less similar. We do the same as in the second experiment, which is actually implemented in the second experiment, and we analyze the results of Table 4. We select 20 graphs in library B1 and different versions of the same graph in library B2 and B3, where library B1 and library B3 have the longest time span, and library B2 is sandwiched between the two. When $\mu = 20$, $\sigma = 0.05$, Fig. 6(b) shows the similarity between different versions of the plot over different time spans, we find that for the same business process graph, whether it is *Structure-Sim* or *Behavior-Sim* or *Comp-Sim*, the similarity between library B1 and library B3 is less than the similarity between library B1 and library B2 or between library B2 and library B3.



(a) Different graphs vs. different versions

(b) different versions

Fig. 6. When $\mu = 20$, $\sigma = 0.05$, the similarity between different graphs and the similarity between different versions of the same graph

5.3 Ablation Experiments

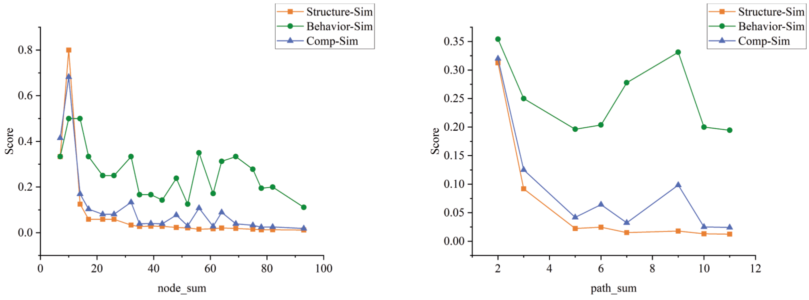
In this part, we discuss in detail the determination of parameters α and β in equations (8) and (9). We also discuss the influence of the number of nodes and paths on the experimental results.

The Influence of Parameter Settings. In our proposed method, we give different weights to structural similarity and behavioral similarity, and we arrive at a relatively comprehensive assessment by balancing the relationship between the two. In the settings, if $\alpha = 0$, the evaluation only considers behavioral similarity, whereas if $\beta = 0$, the evaluation considers only structural similarity.

In the experiment, we set $\mu = 20$, that is selecting 20 business process graphs in libraries A, B3 and C. We calculate the comprehensive evaluation similarity of $\alpha = 0$, $\beta = 1$ and $\alpha = 1$, $\beta = 0$ and $\alpha = 0.5$, $\beta = 0.5$, and obtain the results as shown in Table 5. In our dataset, due to the large number of nodes and the relatively small number of paths, the *Structure-Sim* is relatively small, and the *Behavior-Sim* is relatively large, in this case, if we assign the weight of both metrics to 0.5, the final result will be closer to the behavioral similarity, that is, the result is dominated by behavioral similarity, which loses the meaning of behavioral similarity. However, using our method, the metrics we end up with are not dominated by behavioral similarity, so our weighting method is more comprehensive and objective to the final result.

Table 5. The effect of different parameter settings on the experimental structure

	A-A	A-B3	A-C	B3-B3	B3-C	C-C
$\alpha = 0$,$\beta = 1$	0.2363	0.2269	0.2292	0.3464	0.2969	0.3431
$\alpha = 1$,$\beta = 0$	0.0686	0.0265	0.0249	0.0952	0.0416	0.2407
$\alpha = 0.5$,$\beta = 0.5$	0.1802	0.1593	0.1599	0.2593	0.2072	0.3209
ours	**0.0865**	**0.0479**	**0.0667**	**0.1208**	**0.0893**	**0.2680**



(a) node sum

(b) path sum

Fig. 7. The influence of the number of nodes and the number of paths on the similarity of the graphs

The Influence of the Number of Nodes and Paths. We explore the impact of different number of task nodes and different number of paths on the result. Since different groups of experiments have similar results, we select one of them for visual analysis, when $\mu = 20$, $\sigma = 0.05$, we calculate *Structure-Sim*, *Behavior-Sim* and final *Comp-Sim* between library B1 and library B3, and plot the transformation of each metric with the total number of nodes and the total number of basic paths. For the case where there is the same total number of nodes and basic paths for different pairs of graphs, we calculate the average of all pairs of graphs under the same total. It can be seen from the Fig. 7 that the number of nodes and the number of paths are roughly negatively correlated with *Structure-Sim*. This is because when the structure of the graph is more complex, more editing operations are required to achieve the conversion between the two graphs, which leads to The graph edit distance increases exponentially, so the *Structure-Sim* value will also become smaller. However, there is no absolute linear relationship between the number of nodes and the number of paths on the value of *Behavior-Sim*, the behavioral similarity of two graphs is more correlated with the semantics information of path execution. Since the final result is a comprehensive evaluation of *Structure-Sim* and *Behavior-Sim*, *Comp-Sim* integrates the information of the two, and basically decreases with the increase of the number of nodes and the number of paths, but there are individual points that are affected by *Behavior-Sim* and deviate.

6 Conclusion

In this paper, in order to solve the problem that the isomorphic abstraction simplification of the process graph model makes the original control routing semantics lost, and the original comprehensive evaluation method cannot reflect the real multidimensional space. We proposed a comprehensive evaluation method for business process similarity, which combines the multidimensional similarity features of the business process graph through the TOPSIS method, and uses the entropy weight method on a specific dataset to objectively empower the metrics of the two dimensions, eliminating the subjective judgment of artificial. In order to verify the validity of the evaluation metric, we boldly make three assumptions and explain the rationality of our method by designing a cross-domain and multi-dimension experimental protocol. Through experiments, we verified our assumptions and improved the shortcoming of the traditional use of GED lack of behavioral semantics information.

Acknowledgments. This work was partially supported by the National Natural Science Foundation of China (Grants No. 62276233, 62102366), Key Research Project of Zhejiang Province (2023C01048) and the Natural Science Foundation of Zhejiang Province (Grant No. LQ22F020010).

References

1. Weber, P., Gabriel, R., Lux, T., Menke, K.: Business process management. In: Basics in Business Informatics, pp. 175–206 (2022)
2. Reijers, H.A.: Business process management: the evolution of a discipline. Comput. Indust. **126**, 103404 (2021)
3. Dijkman, R., Dumas, M., Van Dongen, B., Käärik, R., Mendling, J.: Similarity of business process models: metrics and evaluation. Inf. Syst. **36**(2), 498–516 (2011)
4. Thaler, T., Schoknecht, A., Fettke, P., Oberweis, A., Laue, R.: A comparative analysis of business process model similarity measures. In: Dumas, M., Fantinato, M. (eds.) BPM 2016. LNBIP, vol. 281, pp. 310–322. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-58457-7_23
5. Schoknecht, A., Thaler, T., Fettke, P., Oberweis, A., Laue, R.: Similarity of business process models-a state-of-the-art analysis. ACM Comput. Surv. **50**(4), 1–33 (2017)
6. Liu, C., Zeng, Q., Cheng, L., Duan, H., Cheng, J.: Measuring similarity for data-aware business processes. IEEE Trans. Autom. Sci. Eng. **19**(2), 1070–1082 (2021)
7. Cao, B., Wang, J., Fan, J., Dong, T., Yin, J.: Mapping elements with the Hungarian algorithm: an efficient method for querying business process models. In: IEEE International Conference on Web Services, pp. 129–136 (2015)
8. Li, J., Wen, L.J., Wang, J.M.: Process model storage mechanism based on petri net edit distance. Comput. Integr. Manuf. Syst. **19**(8), 1832–1841 (2013)
9. Sebu, M.L., Ciocârlie, H.: Similarity of business process models in a modular design. In: IEEE 11th International Symposium on Applied Computational Intelligence and Informatics, pp. 31–36 (2016)
10. Gao, X., Xiao, B., Tao, D., Li, X.: A survey of graph edit distance. Pattern Anal. Appl. **13**, 113–129 (2010)
11. Pamungkas, E.W., Sarno, R., Munif, A.: Performance improvement of business process similarity calculation using word sense disambiguation. IPTEK J. Proc. Series, 2(1) (2016)
12. Akkiraju, R., Ivan, A.: Discovering business process similarities: an empirical study with SAP Best practice business processes. In: Maglio, P.P., Weske, M., Yang, J., Fantinato, M. (eds.) ICSOC 2010. LNCS, vol. 6470, pp. 515–526. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-17358-5_35
13. Cao, B., Hong, F., Wang, J., Fan, J., Lv, M.: Workflow difference detection based on basis paths. Eng. Appl. Artif. Intell. **81**, 420–427 (2019)

14. Wang, Z.X., Wen, L.J., Wang, S.H., Wang, J.M.: Similarity measurement for process models based on transition-labeled graph edit distance. Comput. Integr. Manuf. Syst. **22**(2), 343–352 (2016)
15. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Philip, S.Y.: A comprehensive survey on graph neural networks. IEEE Trans. Neural Netw. Learn. Syst. **32**(1), 4–24 (2020)
16. Bai, Y., Ding, H., Bian, S., Chen, T., Sun, Y., Wang, W.: Simgnn: a neural network approach to fast graph similarity computation. In: Proceedings of the Twelfth ACM International Conference On Web Search And Data Mining, pp. 384–392 (2019)
17. Li, Y., Gu, C., Dullien, T., Vinyals, O., Kohli, P.: Graph matching networks for learning the similarity of graph structured objects. In: International Conference on Machine Learning, pp. 3835–3845 (2019)
18. Sun, Y., Han, J.: Mining heterogeneous information networks: a structural analysis approach. ACM SIGKDD Explorations Newsl **14**(2), 20–28 (2013)
19. Wang, S., et al.: Heterogeneous graph matching networks for unknown malware detection. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence, pp. 3762–3770 (2019)
20. Aisyah, K.N., Sungkono, K.R., Sarno, R.: A new similarity method based on weighted-linear temporal logic tree and weighted directed acyclic graph for graph-based business process models. Int. J. Intell. Eng. Syst. 13(5) (2020)
21. Zhou, C., Liu, C., Zeng, Q., Lin, Z., Duan, H.: A comprehensive process similarity measure based on models and logs. IEEE Access **7**, 69257–69273 (2019)
22. Zhu, Y., Tian, D., Yan, F.: Effectiveness of entropy weight method in decision-making. Math. Probl. Eng. **1–5**, 2020 (2020)
23. Yoon, K., Hwang, C.L.: Topsis (technique for order preference by similarity to ideal solution)-a multiple attribute decision making, w: Multiple attribute decision making-methods and applications, a state-of-the-at survey. Berlin: Springer Verlag, **128** 140 (1981)
24. Fahland, D., Favre, C., Jobstmann, B., Koehler, J., Lohmann, N., Völzer, H., Wolf, K.: Instantaneous soundness checking of industrial business process models. In: Dayal, U., Eder, J., Koehler, J., Reijers, H.A. (eds.) BPM 2009. LNCS, vol. 5701, pp. 278–293. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03848-8_19
25. Wang, Y., Huaibo Sun, Yu., Zhao, W.Z., Zhu, S.: A heterogeneous graph embedding framework for location-based social network analysis in smart cities. IEEE Trans. Industr. Inf. **16**(4), 2747–2755 (2019)
26. Xiao Wang, Deyu Bo, Chuan Shi, Shaohua Fan, Yanfang Ye, and S Yu Philip. A survey on heterogeneous graph embedding: methods, techniques, applications and sources. IEEE Transactions on Big Data, 2022
27. Eshuis, R., Wieringa, R.: Comparing petri net and activity diagram variants for workflow modelling – a quest for reactive petri nets. In: Ehrig, H., Reisig, W., Rozenberg, G., Weber, H. (eds.) Petri Net Technology for Communication-Based Systems. LNCS, vol. 2472, pp. 321–351. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-40022-6_16
28. Owen, M., Raj, J.: BPMN and business process management. Introduction to the new business process modeling standard, pp. 1–27 (2003)
29. Peterson, J.L.: Petri nets. ACM Comput. Surv. **9**(3), 223–252 (1977)
30. Westfall, L.: The certified software quality engineer handbook. Quality Press (2016)
31. Watson, A.H., Wallace, D.R., McCabe, T.J.: Structured testing: a testing methodology using the cyclomatic complexity metric. NIST Special Public. **500**, 235 (1996)

32. Botman, P.: Testing object-oriented systems: models, patterns and tools. Softw. Qual. Profess. **4**(1), 47 (2001)
33. Agarwal, S., Godboley, S., Krishna, P.R.: Cyclomatic complexity analysis for smart contract using control flow graph. In: Computing, Communication and Learning: First International Conference, pp. 65–78 (2023)
34. Kuhn, H.W.: The Hungarian method for the assignment problem. Naval research logistics quarterly, **2**(1–2), 83–97 (1955)
35. Asratian, A.S., Denley, T.M., Häggkvist, R.: Bipartite graphs and their applications. Cambridge University Press (1998)