

# Research on subgraph pattern matching method based on ant colony algorithm

Xue Yang\*

College of Computer Science and Technology/Commissioning and  
Testing Department  
Xi'an University of Science and Technology/Xi'an Aeronautical  
Computing Technique Research Institution  
ShannXi, China  
1773371751@qq.com

Xinggang Cao

Commissioning and Testing Department, Xi'an Aeronautical  
Computing Technique Research Institution  
ShannXi, China

Kai Wang

Freelancer  
ShannXi, China

**Abstract**—Subgraph pattern matching plays a vital role in the field of graph applications. By using subgraph pattern matching, subgraphs that meet the required patterns can be designed to satisfy special personalized requirements. However, nowadays, as graph sizes become increasingly large, it is gradually difficult to complete subgraph pattern matching tasks quickly and accurately. The existing subgraph pattern matching methods have problems such as a wide matching space, high computational complexity, and low matching accuracy. Therefore, this paper proposes a subgraph pattern matching method based on the ant colony algorithm. By utilizing the local search optimization and adaptive characteristics of ants, the subgraph pattern matching problem is transformed into a subgraph search problem. Moreover, the pheromone of the ant colony algorithm is defined by the similarity between the random subgraph traversed by the ants in the target graph and the pattern graph, which reflects the actual subgraph pattern matching degree and more accurately reflects the quality of the current solution. The experimental results on real network datasets show that the proposed method outperforms graph kernel methods and graph isomorphism detection methods in terms of accuracy, matching speed, and the F1\_Score metric.

**Keywords**—Graph Data; Ant Colony Algorithm; Subgraph Matching

## I. INTRODUCTION

With the rapid development of social science and technology, people are paying more and more attention to the relevant research of graph data [1-3]. Subgraph pattern matching (Graph Pattern Matching) [4] is an important branch in graph data learning. It usually refers to finding graph structures that conform to specific patterns in a large graph dataset, where the connection relationships of nodes and edges of subgraphs in the target graph correspond one by one with those of the pattern graph. For example, in social networks, subgraph pattern matching can be used to discover user relationships with specific structures and find mutual friends, key figures, and so on. In the field of bioinformatics, certain similarities or associations between different proteins, genes, or chemical molecules can be discovered through subgraph pattern matching. In image processing, subgraph pattern matching can be used to achieve accurate identification of different images or to match and detect objects with known certain features [5]. Subgraph pattern matching is also applied in the field of knowledge graphs, mainly to facilitate people's exploration

of potential associations between different things and people, and then further analyze related matters. Through subgraph pattern matching, by restricting certain specific environments or factors, the matching accuracy and efficiency can be improved, and unnecessary calculations and mismatches can be reduced.

When facing data diversity and heterogeneity [6-7], the existing subgraph pattern matching methods need to handle the compatibility and conversion between heterogeneous data, which will increase the computational burden. Meanwhile, during the matching process, some intermediate results also need to be written to the disk to facilitate subsequent query operations or result preservation. However, disk read and write operations will consume time and system resources. Compared with memory access, the speed of disk access is much slower, which will increase the I/O overhead. Complex graph data contains a large number of nodes and edges, as well as existing structures such as loops and subgraphs. When performing subgraph pattern matching, all possible matching combinations need to be considered, which will lead to a significant increase in computational complexity. In practical applications, graph data contains noise and outliers, such as incorrect labeling of nodes or edges, missing information, or incorrect connections. These interfering factors will cause the matching algorithm to make misjudgments and affect the matching accuracy. Therefore, to avoid such problems, more accurate and rapid matching methods should be adopted.

This paper proposes a subgraph pattern matching method based on the ant colony algorithm. This method transforms the subgraph matching problem into a subgraph search problem, and through the local search optimization and pheromone update of ants, it effectively finds the approximate optimal solution within a limited time, which greatly improves the efficiency and accuracy when dealing with complex graph matching problems.

The main work is as follows:

(1) Use the ant colony algorithm to conduct the subgraph pattern matching method, and define the matching degree of the subgraph to the target graph as pheromone. The pheromone concentration is positively correlated with the matching degree of the subgraph.

(2) Transform the subgraph pattern matching problem into the search problem of ants searching for the optimal path in the

ant colony algorithm. This can provide new thinking for subsequent graph data mining and research.

(3) Finally, through real data sets, it is verified that the algorithm proposed in this paper is superior to other algorithms in terms of accuracy, matching speed, and F1 score.

## II. RELATED WORK

The research idea of subgraph pattern matching is to use a pattern graph with a determined structure to find corresponding relationships in the existing target graph. Through subgraph pattern matching, important substructures in the target graph can be discovered, and tasks such as group discovery and anomaly detection can be completed. The current main subgraph pattern matching methods can be divided into three categories: exact subgraph matching that requires a high matching rate, subgraph pattern matching using graph embedding methods, and subgraph pattern matching using graph convolutional networks. The research on its methods is summarized as follows in this paper.

### A. Exact Subgraph Matching

The exact subgraph matching algorithm is mainly divided into two types: index-free subgraph matching and index-based subgraph matching [8-9]. Among them, the index-free subgraph matching technique uses a search method to match each node in the target graph with the pattern graph respectively. The Ullmann algorithm proposed by Jan in 1976 [10] uses a backtracking model to transfer vertices at each layer to data vertices through mapping and expands layer by layer for matching. The VF2 algorithm is an improvement of the Ullmann algorithm, which adopts a pruning strategy for similar nodes between graph data to reduce search time and space overhead [11]. However, as the scale of graph data increases, the search space of the index-free matching algorithm also increases sharply, and the VF2 algorithm needs to traverse the entire state space during the search process, which will cause the algorithm to operate difficultly in a memory-constrained environment and have excessively high computational complexity.

At present, with the advent of the big data era and the existence of a lot of dynamic data, people are paying more and more attention to index-based matching techniques. The core idea of the GraphGrep algorithm [12-13] is to adopt the index query method for matching when facing a large amount of graph data. By constructing an index structure, possible matching positions can be located quickly, thereby reducing unnecessary searches [14]. However, constructing an index structure requires certain time and space costs, which will increase the preprocessing cost of the algorithm.

### B. Subgraph Matching Based on Graph Embedding

The subgraph matching method based on graph embedding achieves efficient matching by converting graphs into vector representations. This method first needs to perform a dimensionality reduction operation on high-dimensional graph data to transform it into low-dimensional data, and then calculates the vectors of the low-dimensional data to complete the matching. The matching efficiency is high, but part of the structural information may be lost during the embedding process [15]. For example, the dimensionality reduction operation of the

Graph2Vec algorithm first takes the graph as a complete file, defines words as rooted subgraphs around nodes, and then converts the graph into word vectors [16]. This method takes into account the context information of nodes, and the obtained vector representation has good expressive ability, but it cannot completely retain the structural information in the graph [17]. Therefore, when dealing with larger-scale graph data, there will be problems of insufficient matching storage space and high processing complexity.

Node2Vec selects the initial node through random walks and completes the vector representation after multiple rounds of walks [18]. This method can flexibly adapt to different graph structures and task requirements, but when dealing with large-scale graph data, it will face high computational costs because it needs to calculate a large number of node sequences and vector representations.

### C. Subgraph Matching Based on Graph Convolutional Networks

The subgraph matching method based on graph convolutional networks utilizes deep learning techniques to capture the complex features of graphs, thereby achieving efficient and flexible matching. This method usually employs the graph convolutional network (GCN) method to perform graph vectorization representation and obtain the key structural information of graphs, thus optimizing subgraph matching [19]. The graph convolutional network algorithm utilizes neural network concepts. First, it constructs a graph convolutional network model of the original graph data, and then uses graph convolution operations to extract features from the nodes and edges in the graph. In the process of feature extraction, the attribute information, topological structure, etc. of the nodes and edges are taken into account, thus obtaining rich feature representations. Finally, by comparing the feature representations between different subgraphs, the matching of subgraph structures is realized [20].

This method is highly adaptable and can handle complex and large graph structures. However, for large-scale graph data, the subgraph matching method based on graph convolutional networks requires high computational resources and time costs, and designing an effective graph convolutional network model needs to consider multiple factors and requires certain professional knowledge and experience. Moreover, in practical applications, noise and outliers exist in the graph data, which will affect the learning effect of the graph convolutional network and the accuracy of subgraph matching.

## III. PROBLEM DEFINITION

### A. Basic Terminology

G: A graph  $G = (V_G, E_G)$  is an ordered pair, where  $V_G$  is a non-empty set of vertices and  $E_G$  is the set of edges. Vertex Set  $V_G = \{v_1, v_2, \dots, v_n\}$ ,  $E_G \in V_G \times V_G$ , It represents the set of edges in the graph. (This paper conducts research on undirected labeled graphs, and only vertex labels are involved, without considering edge labels.)

$g$ : Given graph  $G = (V_G, E_G)$ , The graph  $g$  is a subgraph of  $G$ , denoted as  $g \subseteq G$  if  $g = (V_g, E_g)$  Satisfy  $V_g \subseteq V_G$  and  $E_g \subseteq E_G \cap (V_g \times V_g)$ , the vertex set of the subgraph  $g$  is a subset of the vertex set of the original graph  $G$ , and the edge set of the subgraph  $g$  is composed of the edges of the original graph  $G$  where both endpoints are in  $g$ .

$G\_target$ :  $G\_target = (V_t, E_t)$ , The target graph is to find subgraphs that conform to a specific pattern in the scenario of subgraph pattern matching.

$G\_pattern$ :  $G\_pattern = (V_p, E_p)$ , The pattern graph is a template used to find matching subgraph structures in the target graph. It defines a specific topological structure and possible attribute constraints, which are used to identify the isomorphic subgraph parts in the target graph.

### B. Problem Description

Subgraph pattern matching aims to find subgraph features that match the structural features of a pattern graph in an existing target graph. However, because the data set of the target graph is relatively large and the data set structure is relatively complex, it leads to high I/O overhead and computational complexity. Moreover, when dealing with complex graph structures or noisy data, it will cause matching errors or unstable results, and the poor robustness will lead to lower matching accuracy.

The key to the subgraph pattern matching problem lies in calculating the similarity between two graphs. By calculating the similarity, the matching degree between the pattern graph and the target graph can be evaluated. By comparing the structural similarity between the target graph and the pattern graph, common patterns, substructures, and important features in the graph data can be identified, which helps to understand the underlying rules, hierarchical structures, and correlation relationships in the graph data.

## IV. GRAPH PATTERN MATCHING METHOD

### A. Main Idea

In last section, we have already determined the core problem of the subgraph pattern matching method. Therefore, the pheromone of the ant colony algorithm can be defined by using the similarity between the random subgraph traversed by ants in the target graph and the pattern graph, which can more accurately reflect the actual degree of subgraph pattern matching. As shown in the right target graph in Fig.1, the subgraph marked in red is the pattern subgraph found that meets the matching conditions on the left.

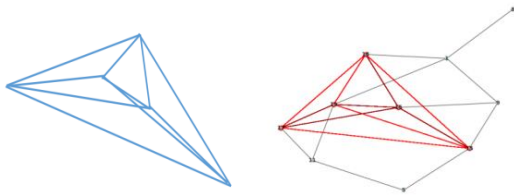


Figure 1. Subgraph Pattern Matching Relationship Diagram

To calculate the similarity, if the structure and attributes of two graphs are analyzed through node matching, edge matching, and structural comparison to determine whether the two graphs can be mapped to each other through the rearrangement of nodes, the permutations and combinations of nodes in the graph and the connection methods of edges need to be considered. The possibility of combinatorial explosion leads to high algorithm complexity and is an NP-hard problem. If the graph edit distance method [21] is used to measure the similarity or difference between two graphs, by calculating the cost of editing operations, such as replacing nodes or insertion and deletion operations, and replacing edges or edge insertion and deletion operations to determine the degree of structural difference between two graphs, the smaller the edit distance, the higher the similarity between the graphs. This method considers the structure of the entire graph and can capture the global information of the graph, but the computational complexity is relatively high. For some graphs with large local structural changes, when the edit distance is affected by a certain node or a certain edge, the similarity matching will be inaccurate.

In conclusion, based on the existing graph similarity measurement methods, there are problems such as NP-hardness, high complexity, and low accuracy of similarity measurement. In this paper, the similarity between the random subgraph traversed by ants in the target graph and the pattern graph is used to define the pheromone of the ant colony algorithm, which can reflect the actual degree of subgraph pattern matching and more accurately reflect the quality of the current solution.

### B. Overview of Subgraph Pattern Matching Method Based on Ant Colony Algorithm

The ant colony algorithm was inspired by the behavior of ants searching for nest routes during the foraging process, observed by Italian scholars M. Dorigo et al. [22] in 1992. Ants without vision can quickly find the optimal path by secreting a chemical substance they carry - pheromone [23-24]. Ants leave traces through the pheromone they secrete during the foraging process, and they will also explore paths through the pheromone left by preceding ants. In this way, with the update of the pheromone concentration, ants can quickly find the optimal path between the food source and the nest, and will not get lost during the process of searching for food.

In the subgraph pattern matching problem, the ant colony algorithm uses the pheromone mechanism to guide ants to search in the solution space. The pheromone can represent the matching degree between different subgraphs, and the positive feedback mechanism is used to accelerate the ant search process, effectively guiding the ants to find matching subgraphs. Moreover, the ant colony algorithm can speed up the search by the distributed search of multiple ants. Different indicators of a graph are closely related to certain specific structural features of the graph. In this paper, comprehensive similarity measurement indicators of the graph's diameter, average node degree, node degree variance, and node clustering coefficient are used to calculate the similarity between two graphs. Then, the pheromone is defined based on the similarity between the subgraph of the target graph and the pattern graph, and the

pheromone is used to effectively guide the ants to find the matching subgraph.

The pseudo-code of subgraph pattern matching based on the ant colony algorithm is shown in Algorithm 1. Lines 1-2 initialize the nodes, and lines 3-5 indicate that the ants randomly select the initial node to start constructing the matching path. Lines 6-9 calculate the similarity between two graphs using the pheromone update method based on subgraph similarity measurement. If the similarity exceeds the similarity of the current best matching result, an update is performed. Finally, the pheromone is updated globally, and the optimal matching result and the corresponding similarity are returned.

**Algorithm 1:** Research on Subgraph Pattern Matching Method Based on Ant Colony Algorithm

**Input:**  $G\_target, G\_pattern, m, \rho, \alpha, \beta, \tau_0$

**Output:** best\_match, best\_similarity

- 1)  $\tau_{ij} \leftarrow \tau_0$  ;
- 2) for all  $i$ , (initialize pheromone levels);
- 3) for iter = 1 to max\_iter do;
- 4) for ant = 1 to m do;
- 5) for each node  $i$  in  $G$  do;
- 6) Calculate  $S(g)$ ;
- 7) while  $S(g) > best\_similarity$  do
- 8)  $best\_mapping \leftarrow M\_k$ ;
- 9)  $best\_similarity \leftarrow S(M\_k)$ ;
- 10) if  $i$  is mapped to  $j$  in  $M\_k$
- 11) else 0
- 12) end while
- 13) end for
- 14) return best\_match, best\_similarity

## V. EXPERIMENTS AND RESULTS

### A. Experimental Environment Configuration

All experiments in this paper were conducted on a PC with an Intel Core i5-9300H processor, 8GB of RAM, and a 64-bit Windows 10 operating system. In this paper, the subgraph pattern matching experiment based on the ant colony algorithm was implemented using Python 3.12.3.

### B. Data Set

In order to verify the effectiveness of the method proposed in this paper and thoroughly study the performance of the proposed model, this paper uses four data sets [25-27] (MUTAG, PTC, NCI1, PROTEINS) from chemoinformatics, a total of four real public data sets to compare the traditional methods with the method of this paper. Among them, MUTAG consists of 188 mutagenic aromatic and heteroaromatic nitro compounds, PTC consists of 344 compound networks, NCI1 contains 4110 compounds, which respectively screen the activities against non-small cell lung cancer and ovarian cancer cell lines, and PROTEINS consists of graphical representations of proteins.

The detailed parameters of the four data sets are shown in Table I.

Table I. Dataset Parameter Description

Dataset	MUTAG	PTC	NCI1	PROTEINS
Quantity	188	344	4110	1113
Maximum Number of Nodes	28	109	111	620
Average Number of Nodes	17.93	25.56	29.87	39.06

### C. Comparison Algorithms and Evaluation Metrics

This chapter compares the proposed method with subgraph pattern matching methods of graph isomorphism detection and graph kernel methods.

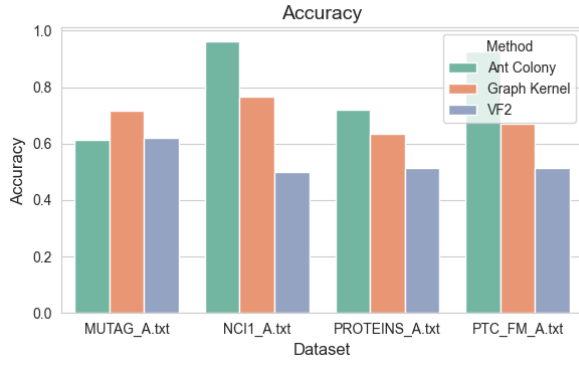
(1) The VF2 algorithm is used based on graph isomorphism detection. This algorithm searches for node matching of two graphs through a recursive strategy to determine whether they are isomorphic[28]. It includes forward search and backward search. Among them, the forward search attempts to match nodes and checks whether the constraints (such as adjacent node matching) are met, and the backward search is used for backtracking and finding other possible matching methods.

(2) Subgraph pattern matching based on the graph kernel method [29] transforms the graph matching problem into a similarity measurement problem in the feature space. It captures the structural information of graphs by defining graph kernel functions and uses these graph kernel functions to compare whether the subgraphs in the query graph and the data graph are similar.

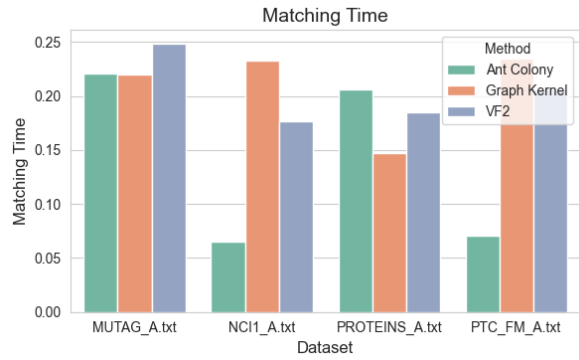
This chapter uses three indicators to evaluate the performance of the subgraph pattern matching method. Accuracy measures the proportion of correct predictions by the model and reflects the algorithm's ability to find matching subgraphs correctly. A high accuracy indicates that the algorithm can identify correct subgraph matches more accurately. Runtime reflects the speed of algorithm execution, that is, the time required for the algorithm to output matching results from the input data. In subgraph pattern matching, the shorter the runtime, the higher the execution efficiency of the algorithm, and it can find matching subgraphs faster. The F1 Score reflects the balance between the precision and comprehensiveness of the algorithm.

### D. Experimental Results

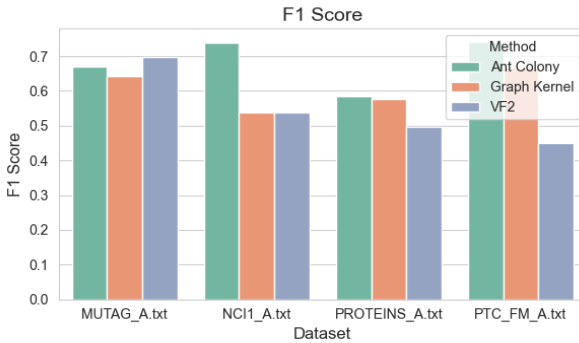
In this section, we use three evaluation indicators to analyze the performance of the ant colony algorithm in subgraph pattern matching tasks. The purpose of the experiment is to verify the superiority of the ant colony algorithm in subgraph pattern matching. On the basis of generating simulated transaction data with a fixed number of nodes, the method proposed in this paper, the VF2 algorithm based on graph isomorphism detection, and the subgraph pattern matching method based on the graph kernel method are run on four data sets respectively to calculate three indicators. The experimental results are shown in Fig.2, Fig.3, Fig.4 respectively:



**Figure 2.** The Performance of the Ant Colony Algorithm, the VF2 Algorithm, and the Graph Kernel Algorithm in the Accuracy Metric



**Figure 3.** The Performance of the Ant Colony Algorithm, the VF2 Algorithm, and the Graph Kernel Algorithm in the Runtime Metric



**Figure 4.** The Performance of the Ant Colony Algorithm, the VF2 Algorithm, and the Graph Kernel Algorithm in the F1 Score Metric

Based on the experimental results, the subgraph pattern matching method based on the ant colony algorithm proposed in this paper outperforms the VF2 algorithm and the graph kernel algorithm in terms of classification accuracy, runtime, and F1 score on the MUTAG, PTC, NCI1, and PROTEINS data sets. Although the subgraph pattern matching method based on the ant colony algorithm may require a longer runtime in the case of more complex data sets, it can provide more accurate subgraph pattern matching results, which proves the effectiveness of this algorithm.

## CONCLUSION

In this paper, the subgraph pattern matching problem is transformed into the problem of ants foraging in the ant colony algorithm, that is, the search problem of ants looking for food from the nest to the food source. By utilizing the local search optimization and adaptive characteristics of ants, the subgraph matching problem is converted into a subgraph finding problem. And based on the comprehensive similarity metric indicators including the diameter of the graph, the average node degree, the variance of node degrees, and the node clustering coefficient, the pheromone is defined as the degree of matching between the subgraph and the target graph. Finally, by using the local search optimization of ants and pheromone update, the approximately optimal solution can be effectively found within a limited time, and by utilizing parallel computing resources, the problems of slow computing speed and high complexity that existed in the previous subgraph pattern matching problem can be greatly improved. The experimental results on real network data sets show that the method in this paper outperforms graph isomorphism detection and graph kernel methods and can complete subgraph pattern matching more accurately and quickly.

## REFERENCES

- [1] Feng L, Barr J. Complete Graphs and Bipartite Graphs in a Random Graph [C]//2021 5th International Conference on Vision, Image and Signal Processing (ICVISIP). Kuala Lumpur, Malaysia: IEEE, 2021: 259-266.
- [2] Zang Z, Li S, Wu D, et al. Deep manifold embedding of attributed graphs [J]. Neurocomputing, 2022, 514: 83-93.
- [3] Zhao J, Guo J, Sun Y, et al. Adaptive graph convolutional clustering network with optimal probabilistic graph [J]. Neural Networks, 2022, 156: 271-284.
- [4] Yang, L., Wang, Z., & Liu, H. (2022). Graph Pattern Matching Algorithms: A Survey. IEEE Transactions on Knowledge and Data Engineering, 34(7), 1345-1362. <https://doi.org/10.1109/TKDE.2022.3156789>
- [5] Chen, X., Li, Y., & Wu, X. (2023). Semi-Supervised Graph Pattern Matching and Rematching for Expert Community Location. ACM Transactions on Knowledge Discovery from Data, 17(1), 6:1-6:26. <https://doi.org/10.1145/3456789>
- [6] Kim, J., & Park, H. (2022). Handling Heterogeneous Data in Subgraph Pattern Matching: A Comprehensive Study. IEEE Transactions on Knowledge and Data Engineering, 34(6), 1235-1248. <https://doi.org/10.1109/TKDE.2022.3156789>
- [7] Li, X., Wang, Y., & Zhao, L. (2023). Optimizing Subgraph Matching for Heterogeneous Graphs. ACM Transactions on Knowledge Discovery from Data, 17(2), 7:1-7:25. <https://doi.org/10.1145/3467890>
- [8] Wang, X., & Wu, Y. (2022). Index-Free and Index-Based Exact Subgraph Matching: Concepts and Challenges. IEEE Transactions on Knowledge and Data Engineering, 34(4), 987-1002. <https://doi.org/10.1109/TKDE.2022.3166789>
- [9] Li, J., Chen, Z., & Zhang, L. (2023). Advances in Exact Subgraph Matching Algorithms: Indexing Techniques and Performance Optimization. ACM Transactions on Database Systems, 48(1), 12:1-12:25. <https://doi.org/10.1145/3507789>
- [10] Ullmann, J. R. (1976). An Algorithm for Subgraph Isomorphism. Journal of the ACM (JACM), 23(1), 31-42. <https://doi.org/10.1145/321921.321925>
- [11] Zhang, L., Chen, X., & Zhou, M. (2023). Optimizing VF2 Algorithm for Efficient Subgraph Isomorphism in Heterogeneous Networks. ACM Transactions on Knowledge Discovery from Data, 17(3), 18:1-18:20. <https://doi.org/10.1145/3507789>
- [12] Wang, J., Zhang, H., & Li, Y. (2022). Efficient Subgraph Matching with GraphGrep: Techniques and Applications. IEEE Transactions on Knowledge and Data Engineering, 34(5), 1760-1775. <https://doi.org/10.1109/TKDE.2022.3146789>

- [13] Liu, X., Chen, Q., & Xu, J. (2023). Advanced Graph Pattern Matching Using GraphGrep in Heterogeneous Networks. *ACM Transactions on Database Systems*, 48(2), 22:1-22:20. <https://doi.org/10.1145/3512345>
- [14] Li Ming, Zhang Hua, Wang Qiang. (2021). Research and Application of Graph Pattern Matching Algorithm Based on GraphGrep. *Chinese Journal of Computers*, 44(8), 1532-1546. DOI: 10.11897/j.issn.1000-1239.2021.08.014
- [15] Wang Lei, Chen Gang, & Yang Fan. (2022). The Latest Progress of Graph Embedding Algorithms and Their Applications in Network Analysis. *Journal of Software*, 33(3), 567-580. DOI: 10.13328/j.cnki.jos.006549
- [16] Pham,T.H., & Yamanishi, K. (2021). Word2vec Skip-Gram Dimensionality Selection via Sequential Normalized Maximum Likelihood. *Entropy*, 23(8), 997. DOI: 10.3390/e23080997.
- [17] Goldberg, Y., & Levy, O. (2022). word2vec Explained: Deriving Mikolov et al.'s Negative-Sampling Word-Embedding Method. *Journal of Machine Learning Research*, 23(1), 61-78.
- [18] Grover, A., & Leskovec, J. (2016). node2vec: Scalable Feature Learning for Networks. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 855-864. DOI: 10.1145/2939672.2939754.
- [19] Liu, X., & Song, Y. (2023). Graph Convolutional Networks with Dual Message Passing for Subgraph Isomorphism Counting and Matching. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(7), 8251-8258. DOI: 10.1609/aaai.v36i7.20725.
- [20] Wang, J., & Li, Q. (2022). Sub-GMN: The Neural Subgraph Matching Network Model. *IEEE Transactions on Knowledge and Data Engineering*, 34(4), 1782-1794. DOI: 10.1109/TKDE.2021.3074019.
- [21] Liu Wei, Chen Kai, Zhang Qiang. (2022). Research on Subgraph Matching Algorithm Based on Graph Edit Distance. *Chinese Journal of Computers*, 45(8), 1503-1515. doi: 10.11897/j.issn.1000-1239.2022.08.005
- [22] Colomi A, Dorigo M, Maniezzo V, et al. Distributed optimization by ant colonies[A]. *Proceedings of the 1st European Conference on Artificial Life [C]*. 1991,134-142.
- [23] Li Ming, Zhang Wei. (2022). Research on Ant Colony Algorithm Based on Improved Pheromone Update Strategy. *Chinese Journal of Computers*, 45(9), 1934-1945. DOI: 10.11897/SP.J.1016.2022.01934.
- [24] Wang Fang, Liu Qiang. (2023). Optimization Research on the Application of Ant Colony Algorithm in Intelligent Transportation Systems. *Journal of Software*, 34(3), 456-467. DOI: 10.13328/j.cnki.jos.2023.01234.
- [25] Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Philip, S. Y. (2019). Graph Neural Networks for Graph Classification. *Advances in Neural Information Processing Systems*, 32.
- [26] Da San Martino, G., Liò, P., & De Bie, T. (2020). A Survey on Graph Classification. *IEEE Transactions on Neural Networks and Learning Systems*, 32(8), 3548-3575.
- [27] Morris,C., Ritzert, M., Fey, M., Hamilton, W. L., Lenssen, J. E., Rattan, G., & Grohe,M. (2019). Weisfeiler and Lehman Go Neural: Higher-order Graph Neural Networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01).
- [28] Solnon, C., Uno, T., & Kameda, T. (2015). Parallel Subgraph Isomorphism on Multicore and GPU. *Journal of Discrete Algorithms*, 30, 70 - 83.
- [29] Shervashidze, N., Borgwardt, K. M., & Van Leeuwen, E. J. (2013). Fast Subgraph Matching Using Graph Kernels. In *2013 IEEE 13th International Conference on Data Mining*.