frontiers | Frontiers in Big Data

TYPE Original Research
PUBLISHED 10 February 2025
DOI 10.3389/fdata.2025.1546850

Edge-level multi-constraint graph pattern matching with lung cancer knowledge graph

Houdie Tu[1], Lei Li[2,3]*, Zhenchao Tao[4,5] and Zan Zhang[2,3]

[1]School of Artificial Intelligence, Hefei University of Technology, Hefei, China, [2]Key Laboratory of Knowledge Engineering with Big Data (the Ministry of Education of China), Hefei University of Technology, Hefei, China, [3]School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, China, [4]Department of Radiation Oncology, The First Affiliated Hospital of USTC, Division of Life Sciences and Medicine, University of Science and Technology of China, Hefei, China, [5]Department of Radiation Oncology, Anhui Provincial Cancer Hospital, Hefei, China

OPEN ACCESS

EDITED BY
Guanfeng Liu,
Macquarie University, Australia

REVIEWED BY
Peng Shi,
University of Science and Technology Beijing, China
Qixing Qu,
University of International Business and Economics, China
Yi Zhu,
Yangzhou University, China

*CORRESPONDENCE
Lei Li
✉ lilei@hfut.edu.cn

RECEIVED 17 December 2024
ACCEPTED 24 January 2025
PUBLISHED 10 February 2025

CITATION
Tu H, Li L, Tao Z and Zhang Z (2025) Edge-level multi-constraint graph pattern matching with lung cancer knowledge graph. *Front. Big Data* 8:1546850.
doi: 10.3389/fdata.2025.1546850

Introduction: Traditional Graph Pattern Matching (GPM) research mainly focuses on improving the accuracy and efficiency of complex network analysis and fast subgraph retrieval. Despite their ability to return subgraphs quickly and accurately, these methods are limited to their applications without medical data research.

Methods: In order to overcome this limitation, based on the existing research on GPM with the lung cancer knowledge graph, this paper introduces the Monte Carlo method and proposes an edge-level multi-constraint graph pattern matching algorithm TEM with lung cancer knowledge graph. Furthermore, we apply Monte Carlo method to both nodes and edges, and propose a multi-constraint hologram pattern matching algorithm THM with lung cancer knowledge graph.

Results: The experiments have verified the effectiveness and efficiency of TEM algorithm.

Discussion: This method effectively addresses the complexity of uncertainty in lung cancer knowledge graph, and is significantly better than the existing algorithms on efficiency.

KEYWORDS

graph pattern matching, probability graph, lung cancer knowledge graph, Monte Carlo method, multi-constranint

1 Introduction

Graph pattern matching (GPM) has always been crucial in graph computing, evolving to meet the requirements of emerging applications. The field of graph pattern matching was originally rooted in protein isomorphism research (Hu and Ferguson, 2016; Tian and Patel, 2008) and later expanded to cover community discovery (Liu et al., 2021; Su et al., 2022), expert identification (Li et al., 2016; Wei et al., 2014), development of recommender systems (Fan et al., 2013), social group discovery (Khan et al., 2020; Sato et al., 2016; Chikhaoui et al., 2020), and range group identification (Fan et al., 2010). In 2024, Li et al. Li et al. (2024) introduced the concept of probability graph pattern matching for lung cancer knowledge graph, and proposed a multi-constraint graph pattern matching algorithm TKG-McGPM, which combined with Monte Carlo method and candidate node screening to improve the diversity and validity of matching results. However, we believe there are some disadvantages that: (1) Different parameter characteristics: The node trust value T_{DT} is based on explicit relationships or attributes, and it is not suitable for the Monte Carlo method, while the marginal parameter diagnosis and treatment cycle value T_{DC} and the cost-benefit analysis value T_{CV} involve uncertainty factors, and they are more suitable for the Monte Carlo method. (2) Computational complexity and resource consumption: Monte Carlo methods require a lot of computational resources and are suitable for edge matching rather than node matching, as the latter tends to be more efficient and of lower complexity.

Therefore, it is more reasonable and efficient to apply the Monte Carlo method to edge matching. Hence, this paper proposes an edge-level multi-constraint graph pattern matching algorithm (TEM) based on lung cancer knowledge graph, which ensures the correctness of the matching results and increases the diversity by using the random method to obtain the values of the two parameters T_{DC} and T_{CV} on the edge. In order to further discuss the effectiveness and efficiency of using the Monte Carlo method on both nodes and edges, a hologram multi-constraint pattern matching algorithm (THM) has been proposed. Experimental results show that the TEM algorithm is superior to the existing algorithms and the THM algorithm. All in all, contributions of this paper include:

- The necessity of employing Monte Carlo methods for edge matching to achieve superior subgraph matches is proposed;
- To ensure optimal pattern graph alignment, the conventional graph pattern matching model is refined, introducing an edge-level approach tailored specifically to a lung cancer knowledge graph;
- Addressing prevailing challenges, we propose TEM, an edge-level graph pattern matching algorithm grounded in the context of a lung cancer knowledge graph;
- To rigorously assess the efficacy and efficiency of the TEM algorithm, we further introduce THM, a hologram pattern matching algorithm also rooted in the lung cancer knowledge graph framework.

2 Preliminary

2.1 Lung cancer knowledge graph

The knowledge graph used in this paper is derived from the tumor knowledge graph designed by Li et al. (2024) which consists of five participating nodes: attending physician (AP), testing instrument (TI), tumor type (TC), nursing staff (PM) and treatment method (TM). The pattern graph is shown in Figure 1 and the data graph is shown in Figure 2, where each node has an associated trust value reflecting its trustworthiness in the eyes of others.

The relevant concepts are defined as follows:

Definition 2-1: *Diagnostic and therapeutic trust value* T_{DT} represents the degree of trust between participating nodes in the tumor knowledge graph, ranging from 0 to 1.

Definition 2-2: *Diagnosis and treatment cycle value* T_{DC} represents the efficiency of resource allocation among these nodes, also in the range of 0 to 1.

Definitions 2-3: *Diagnostic and therapeutic cost-benefit analysis value* T_{CV} compares the cost of a medical intervention with its outcomes (e.g., survival and quality of life) to determine which intervention provides the highest cost-benefit ratio. There are two ways to calculate it: (1) The first approach (C/E) compares the effect under fixed costs and focuses on cost savings. It calculates the cost per unit of effect, such as the amount spent each year to extend life. This approach emphasizes cost savings. (2) The second approach (E/C) compares the effect of fixed costs and emphasizes the improvement in survival. It determines the effect generated per unit cost, considering the case of how a particular cost prevents multiple infection-related complications. This approach emphasizes improving survival. In this study, we assume that patients prioritize improved survival. Therefore, the patient chooses the second calculation method T_{EC}. We compute T_{DC} and T_{CV} along the path by multiplication and T_{DT} by averaging.

2.2 Pattern graph matching

According to the correspondence between the data graph and the pattern graph, the graph pattern matching study can be divided into two categories: isomorphic GPM and simulated GPM. Isomorphic pattern matching requires a double-shot function to ensure that the topology of the matching subgraph perfectly reflects the pattern graph. Typical algorithms include VF2 (Foggia et al., 2001), VF3 (Carletti et al., 2018), R-join (Cheng et al., 2008), and G-Ray (Tong et al., 2007). This type of matching is key in 3D object matching and protein structure matching, and indexing, parallelization, and distribution methods are often used to improve efficiency. Due to its NP integrity, the computationally high cost makes the strict matching standard unsuitable for applications where accuracy is not a primary consideration. Based on this, scholars turned to simulation-based graph pattern matching. The concept of graph simulation was first proposed by Henzinger et al. (1995), and it requires that the nodes in the matching subgraph maintain the same successor relationship as their corresponding nodes in the pattern graph. Fan et al. (2010) converted exact one-to-one matching into binary relational search by bounded length, and Liu et al. (2015) extended bounded simulation to accommodate multi-constraint graph pattern matching, combining node and edge attribute information. Liu et al. (2020) proposed the multi-fuzzy constraint graph pattern matching to solve the limitation of ignoring the precursor adjacency relationship of the existing model, and introduced the strong simulation matching model of multi-fuzzy constraints, which matched the precursor relationship and the successor relationship of the candidate nodes at the same time, and effectively eliminated the nodes that did not meet the adjacency relationship. In addition, Liu et al. (2022) proposed a graph pattern matching model, which considers the number of nodes matched by each node in a fixed pattern graph, which is especially important when the matching subgraph contains too many matching nodes. For semi-supervised graph pattern matching with multiple constraints, Yan (2023) proposed a semi-supervised graph pattern matching algorithm based on bisimulation edge sequence guidance named DS-ES-SS GPM, which added the preference of decision makers to the matching process. Jin et al. (2023) proposes a strong simulation matching algorithm TPC-GPSSM based on timing priority constraint. The algorithm adds time order constraint in the matching process of the graph topology structure of the pattern graph to achieve the purpose of pruning in advance and reducing the computational complexity. Guo (2024) used graph pattern matching technology to predict the quality problems of the slab caused by the fluctuation

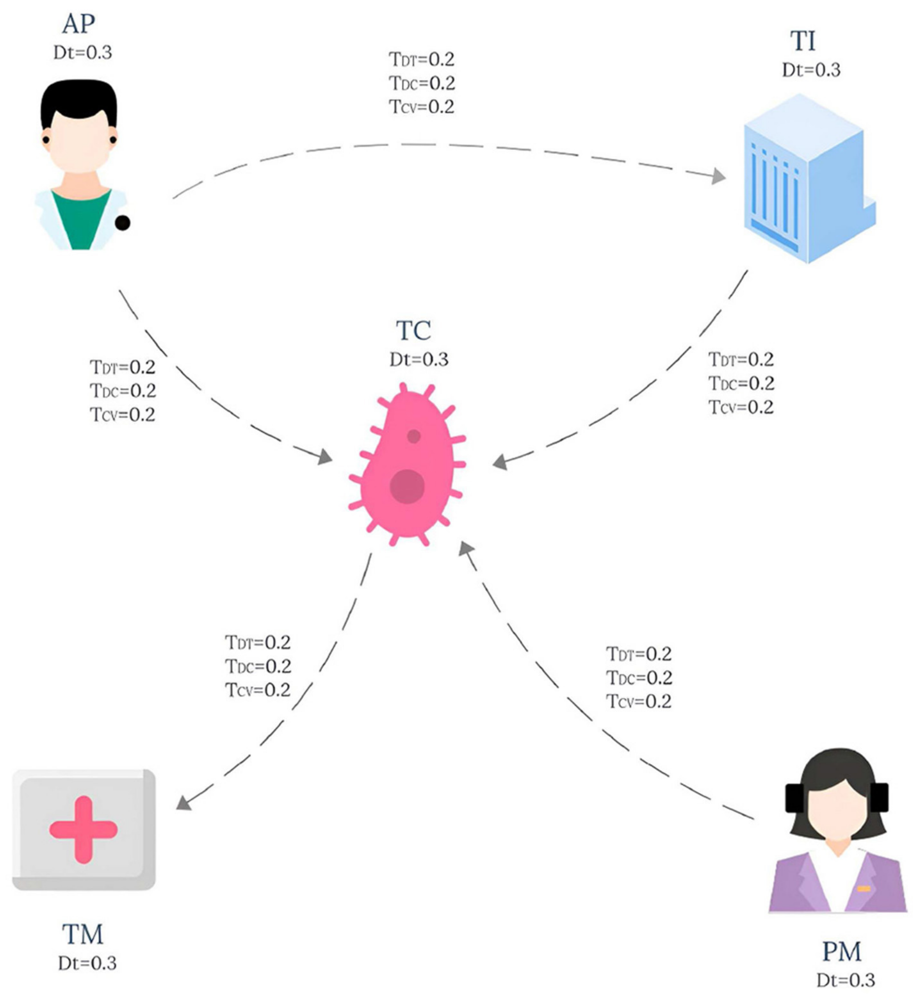

FIGURE 1
Pattern graph.

of the characteristic mode of working conditions in the continuous casting process in real time, so as to provide guidance for the quality improvement of the slab.

In order to study the application effect of graph pattern matching in medical field, Li et al. (2024) introduced the concept of probability graph pattern matching specially applicable to lung cancer knowledge graph, and proposed a multi-constrained graph pattern matching algorithm TKG-McGPM that combines Monte Carlo method and candidate node screening, aiming to enhance the diversity and effectiveness of matching results and assist patients in selecting the best tumor treatment plan. However, although the TKG-McGPM algorithm improves the matching performance, it still faces the challenge of balancing computational efficiency and accuracy when dealing with complex graph structures. Especially in the parameter processing of edges, how to effectively use the Monte Carlo method for optimization has become a key problem. Based on this, we propose the Edge Standardized Monte Carlo Matching Method (EdgeNormMC) to discuss the effect of applying the Monte Carlo method to the two parameters of the edge. In this method, the values of the two parameters T_{DC} and T_{CV} on the edges are scaled to the interval [0,1], and then the Monte Carlo method is used to perform multiple random sampling in the specified interval to select the optimal candidate edges from the possible matching set. The whole matching process is shown in Figure 3.

Definition 2-4: A data graph $G_D = (V, E, f_V^D, f_E^D)$ is a directed graph with node and edge attributes, where:

- V is the set of nodes of the data graph;
- E is the set of edges of the data graph, and $(v_i, v_j) \in E$ represents the directed edge from node $v_i \in V$ to node $v_j \in V$;
- f_V^D is a function defined on a set of nodes, and in a Medical Knowledge Graph (TKG), each node has an attribute constraint value T_{DT} and a label ρ, where ρ represents the type of node, and the value of ρ can be AP, TI, TC, PM and TM;
- f_E^D is a function defined on a set of edges, and $\forall e \in E, f_E^D(v_i, v_j)$ is the attribute set of e. In a TKG, for the directed edge (v_i, v_j), $f_E^D(v_i, v_j)$ contains T_{DT}, T_{DC} and T_{CV}.

Definition 2-5: A pattern graph $G_P = (V_P, E_P, f_V^P, f_E^P, f_l^P, f_m^P)$ is a directed graph with node and edge attributes, where:

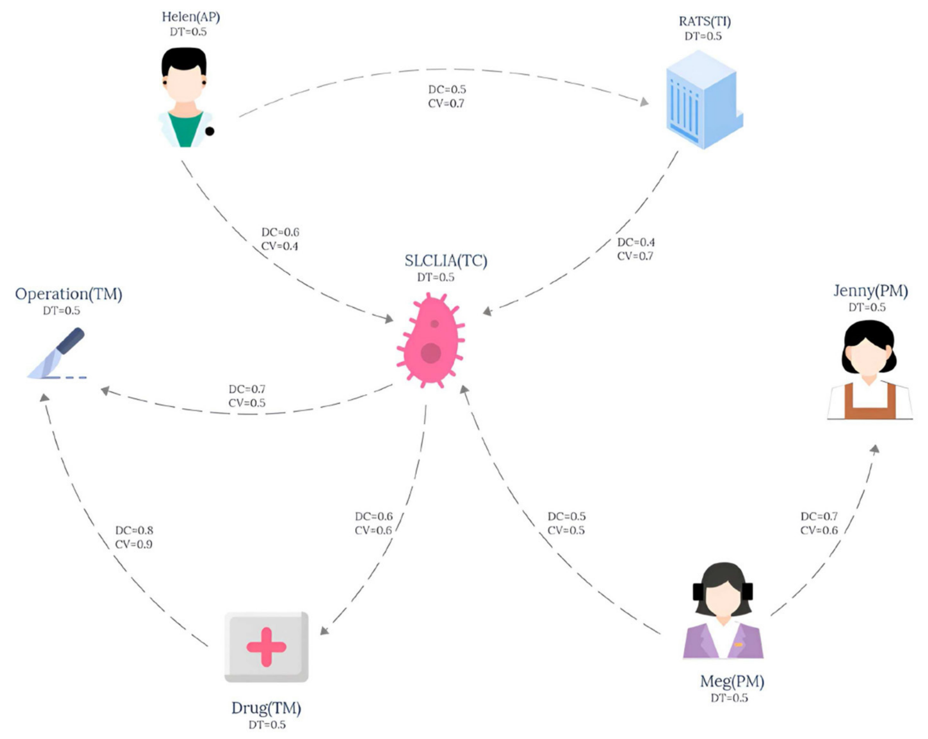

FIGURE 2
Data graph.

- V_P is the set of nodes of the pattern graph;
- E_P is the set of edges of the pattern graph, and $(u_i, u_j) \in E_P$ represents the directed edge from node $u_i \in V_P$ to node $u_j \in V_P$;
- f_V^P is a function defined on V_P, and $\forall u \in V_P$, $f_V^P(u)$ is the attribute set of u. In TKG, the function $f_V^P(u)$ corresponding to node u has the same meaning as the attribute of the node set in the data graph above;
- f_E^P is a function defined on E_P, and $\forall e \in E_P, f_E^P(e)$ is the attribute set of e such that for each edge in E_P, and $f_E^P(u_i, u_j)$ is the set of properties associated with (u_i, u_j).
- f_l^P is a function defined on E_P, and $\forall (u_i, u_j) \in E_P, f_l^P(u_i, u_j)$ is the length constraint of the edge (u_i,u_j), whose values are positive integers k or symbols $*$, respectively, indicating that the interval of length u_i to u_j of the edge does not exceed k or there is no length limit. In TKG, without loss of generality, $f_l^P(u_i, u_j) = 2$.
- f_m^P is a set of membership constraint functions defined on node properties or edge properties.

Definition 2-6 (Edge normalized Monte Carlo matching method, EdgeNormMC): For a pattern graph $G_P = (V_P, E_P, f_V^P, f_E^P, f_l^P, f_m^P)$ and a data graph $G_D = (V, E, f_V^D, f_E^D)$, G_D matches G_P, denoted as $G_P \trianglelefteq G_D$, if there is a binary relationship $S \subseteq V_P \times V$:

- For all $u \in V_P$, there is $v \in V$ such that $(u, v) \in S$;
- For each pair $(u, v) \in S$,
  - If there is a membership calculation function for node attribute in f_m^P, then the corresponding attribute in $f_V^D(v)$ only needs to satisfy the corresponding membership constraint; otherwise, $f_V^D(v)$ needs to satisfy the constraint $f_V^P(u)$ defined on node u;
- $u \sim v, f_V^P(u) = f_V^D(v)$, and
- For each edge (u, u'), from the data graph G_D, there is a path p from v to v', so that $(u', v') \in S, f_V^P(u') = f_V^D(v')$ and if $f_l^P(u, u') = k$, then the length of the interval from node v to node v' in the path $plen(p) \leq k$;
- If there is a membership calculation function for the aggregated attributes on the matching path in f_m^P, the corresponding aggregated attributes in the edge only need to satisfy the corresponding membership constraints; otherwise, the aggregated attributes on the matching path need to satisfy the corresponding attribute constraints in f_E^P.

Example 2.1. Consider a lung cancer diagnosis and treatment plan that needs to be consulted, and the plan needs to be organized by five nodes: Attending physician (AP), Testing instruments (TI), Types of lung cancer (TC), paramedic (PM), and Treatment (TM), and the interaction between them is shown in **Figure 1**. The data graph can be expressed as $G_D = (V, E, f_V^D, f_E^D)$, where represents the role type, role name and trust impact factor T_{DT}, and f_E^D represents the diagnosis and treatment cycle value T_{DC} and the diagnosis and treatment cost-benefit analysis value T_{CV} between participants. The pattern graph can be expressed as $G_P = (V_P, E_P, f_V^P, f_E^P, f_l^P, f_m^P)$, where f_V^P represents the role constraint and trust impact factor constraint T_{Dt} for pattern nodes, f_E^P represents the diagnosis and treatment trust constraint T_{DT}, diagnosis and treatment cycle value constraint T_{DC}, diagnosis and treatment cost

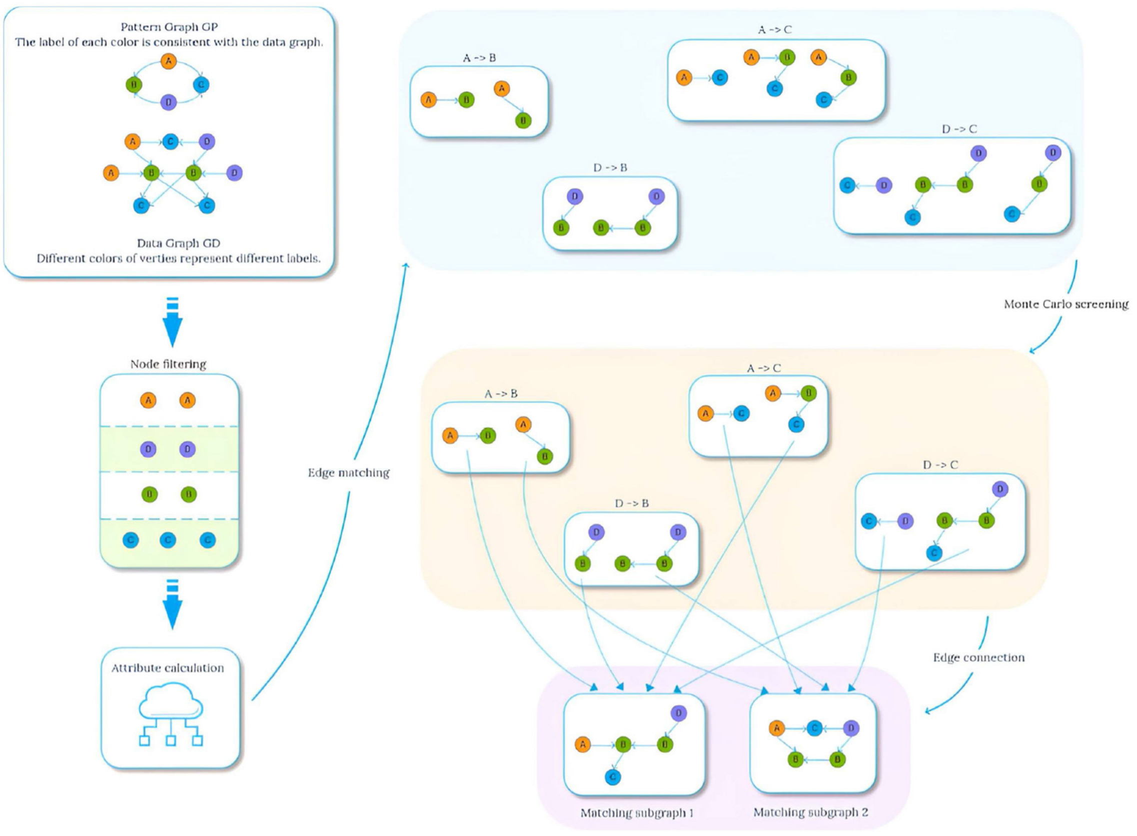

FIGURE 3
Matching process.

benefit analysis value constraint T_{CV} for matching path of pattern edges, and f_l^P represents the matching path length constraint for pattern edges. $f_m^P = (f_{Dt}^m, f_{DT}^m, f_{DC}^m, f_{CV}^m, T_{Dtm}, T_{DTm}, T_{DCm}, T_{CVm})$, where f_{DT}^m represents the membership calculation function defined on the node trust impact factor constraint T_{Dt}, and T_{Dtm} represents the corresponding membership constraint value. f_{DT}^m,f_{DC}^m and f_{CV}^m respectively represent the membership functions defined on the pattern edge attribute constraints T_{DT}, T_{DC} and T_{CV}, while T_{DTm}, T_{DCm} and T_{CVm} respectively represent the membership constraint values of the corresponding attribute constraints.

For the convenience of calculation, The membership constraint values T_{DT}, T_{DC} and T_{CV} of each attribute are set to 0.2, and T_{Dtm} is set to 0.3. In this study, the main purpose of specifying the uniform value of the constraint value is to simplify the calculation process, so that readers can understand the matching calculation process more clearly and smoothly. It should be pointed out that in the actual correlogram, the values corresponding to each edge are not uniform, but show different situations, and these values are randomly generated. According to the EdgeNormMC definition, we can get the matching node Helen of AP in the pattern graph, because the outgoing edge (AP,TC) of AP matches the path(Helen,SLCLIA) in the data graph, and the outgoing edge (AP,TI) matches the path(Helen,RATS) in the data graph. The matching node SLCLIA of TC is determined because we can get that the incoming edge (AP,TC) of TC matches the path(Helen,SLCLIA) in the data graph, the incoming edge (TI,TC) matches the path(RATS,SLCLIA) in the data graph, and the incoming edge (AP,TC) matches the path(Meg,SLCLIA) in the data graph. We can get results similar to those of other matching nodes in the model, the final matched subgraph $G_{sub} = (V_{sub}, E_{sub})$, including V_{sub} = (Helen, RATS, SLCLIA, Operation, Drug, Meg). E_{sub}=(Helen,RATS), (Helen,SLCLIA), (RATS,SLCLIA), (SLCLIA,Operation), (SLCLIA,Drug), (Drug,Operation), (Meg,SLCLIA).

3 GPM with lung cancer knowledge graph

At present, there are two main types of multi-constraint graph pattern matching algorithms. One is composed of two core modules, that is, the matching of pattern edges and the connection of matching paths based on the topology of the pattern graph. The other is based on sequential exploration of the topology of pattern nodes. The existing graph pattern matching algorithm in lung cancer domain, TKG-McGPM, adopts the matching algorithm NTSS based on topological ordered exploration of pattern nodes, and introduces Monte Carlo method in the node matching process, which has achieved certain results. However, considering that node matching focuses more on certainty and accuracy, while edge matching is more suitable for dealing with uncertainty and probabilistic simulation optimization, we believe that Monte Carlo method applied to edge matching may be more reasonable and efficient.

To solve the above problems, we first propose an edge-level graph pattern matching algorithm TEM based on lung cancer knowledge graph by using Monte Carlo method on edges. Experimental results show that the proposed algorithm is significantly better than the existing TKG-McGPM algorithm in performance. In order to further explore the application potential of Monte Carlo method in lung cancer graph pattern matching, we also propose a hologram pattern matching algorithm THM based on lung cancer knowledge graph. In this section, the main algorithm flow of the TEM algorithm will be detailed, and the THM algorithm will be introduced in the next section.

3.1 Description of the TEM algorithm

The matching process of pattern nodes is divided into two key stages. Firstly, according to the constraints on the pattern nodes in the pattern graph, the candidate nodes that meet the conditions are selected. Then, according to the topological structure characteristics of the pattern nodes in the pattern graph, the selected candidate nodes are further filtered.

```
Require: Node whose mode is to be matched u∈ $V_P$, Node
  set V of the data graph
Ensure: The set of candidate nodes $Cand_u$ of $u$
  while For each node V in v, if V.visited =false do
    if $label_v$(u) ∈ $label_v$(v) and $f^m_D t(Dt)$ ≥ Dtm then
      Add $v$ to $u$'s set of candidate nodes set $Cand_u$
    end if
  end while
  return $Cand_u$
```

Algorithm 1. GetNodeCandidate Algorithm for pseudo code

In the lung cancer knowledge graph matching described in this paper, the constraints on the nodes include the constraint $label_v$ on the node label and the constraint T_{Dt} on the node trust impact factor. In addition, f^P_m also contains a membership calculation function f^m_{Dt} for node trust impact factor constraint T_{Dt} and the corresponding membership constraint T_{Dtm}. For a pattern node u, we have $label_v(\mathrm{u}) \subset label_v(v)$ and $f^m_{Dt}(Dt) \geq T_{Dtm}$ if there exists a candidate node $v \in V$, and the set $Cand_u$ of candidate nodes of u can be obtained by the GetNodeCandidate method as shown in Algorithm 1.

The TEM algorithm proposed in this paper still uses the node topology order to match pattern nodes, and this method can prune invalid matches by judging whether the candidate nodes meet the previous topology structure.

The input of the TEM algorithm is data graph G_D and pattern graph G_P, and the output is the set G^{All}_{sub} of matching subgraphs. Firstly, a topologically ordered sequence V_T of pattern nodes and a node V_E with indegree 0 are obtained by topological sorting algorithm. Then the GetNodeCandidate algorithm is called to obtain the candidate node $Cand_{us}$ of the starting matching node u_s, and the above steps are shown in lines 1-2 of Algorithm 2.

```
Require: Data graph $G_D$ and Pattern graph $G_P$
Ensure: Set of all matching subgraphs $G^{All}_{sub}$
  Get the topological sorting sequence $V_T$ of the
  pattern graph and the node $V_E$ with zero degree
  Call GetNodeCandidate algorithm to get candidate
  node $Cand_{us}$, which starts with matching node $u_s$
  Example Initialize the storefile,i = 0, num=0
  while i< length($Cand_{us}$) do
    $v_s$=$Cand_{us}$[i]
    Edge($v_s$,v')=EdgeMatching($v_s$,($u_s$,u'))
    $G_{temp}$=EdgeAttribute(Edge($v_s$,v'))
    if There is a node corresponding to the start
  node of the pattern edge then
      $G_{sub}$=Recursivematching(num+1,$V_T$,$V_E$,storefile)
    end if
    Add the intermediate result $G_{sub}$ to $G^{All}_{sub}$
    i=i+1
  end while
  return $G^{All}_{sub}$
```

Algorithm 2. TEM algorithm

It loops through the candidate node set of the starting node, and then uses the pattern edge matching methods EdgeMatching and EdgeAttribute to match all matching paths that meet the conditions, and then calls the Recursivematching method to obtain the matching subgraph G_{sub}, and adds the matching subgraph to the matching subgraph set G^{All}_{sub}. This is shown in lines 4-10 of Algorithm 2.

The specific execution steps of the Recursivematching method are shown in Algorithm 3. Firstly, it is determined whether the number of the currently processed pattern edge is equal to the number of edges of the pattern edge. If it is equal, it means that all pattern edges have been processed and the matching results can be judged and stored, as shown in lines 1-5 of Algorithm 3. If not, loop through the topological sort nodes and match the pattern edges for each candidate node, as shown in lines 7-16 of Algorithm 3. The result that has been matched is read from the cache, and if there is no one in the cache, the GetNodeCandidate function is called to obtain the candidate node set $Cand_{uc}$ of u_c, as shown in lines 9-12 of Algorithm 3. For each candidate node v_c, EdgeMatching and EdgeAttribute are called to filter edges with multiple constraints and match the next edge recursively, as shown in lines 13-16 of Algorithm 3.

Example 3.1. Taking the pattern graph in Figure 1 and the data graph in Figure 2 as examples, the TEM algorithm firstly sorts the nodes in the pattern graph shown in Figure 1, and obtains the topologically ordered sequence V_T ={AP,PM,TC,TI,TM} of pattern nodes and the set V_E ={AP,PM} of pattern nodes with in-degree 0. Then, the set of candidate nodes $Cand_{AP}$={Helen} of AP is obtained. Helen is added to the list of AP candidate nodes in G_{temp}, and the pattern edge (AP,TI) matching starts from Helen. After the matching path (Helen,RATS) is obtained, the candidate node set $Cand_{TI}$={RATS} of pattern node TI is also obtained. Then, starting from the second node of the pattern node topological sorting

```
Require: The number of the pattern edge num, the
    topological sort sequence $V_T$ and the node $V_E$ with
    zero degree
Ensure: The result storefile
    if num=the number of sides of the pattern graph
    then
        for All nodes x of the pattern graph do
            Clear the matching nodes and edges that do
    not meet the conditions
            Store the matching result $G_{temp}$ to storefile
        end for
        return
    end if
    i=1
    while i<length($V_T$) do
        $u_c$=$V_T$[i]
        if $u_c \in V_E$ then
            Call the GetNodeCandidate function to get the
    candidate node set $Cand_{uc}$ for $u_c$
        elseRead results $Cand_{uc}$ from $G_{temp}$
        end if
        for Each candidate node $v_c$ in $Cand_{us}$ do
            Edge($v_c$,v')=EdgeMatching($v_c$,($u_c$,u'))
            $G_{temp}$=EdgeAttribute(Edge($v_c$,v'))
        end for
        Recursivematching(num+1,$V_T$,$V_E$,storefile)
    end while
    return storefile
```

Algorithm 3. Recursivematching algorithm.

```
Require: Pattern edge (u,u'), start node v∈V and data
    graph $G_D$
Ensure: Set Edge(v,v') that satisfies the path length
    constraint
    Initialization Q = ∅
    while Queue Q is not empty do
        Take a path from Q pathj(v,v')
        if $path_j$(v,v') is not in Edge(v,v') then
            if There is another path from v to v' in
    Edge(v,v') then
                Add $path_j$(v,v') to pathlist(v,v')
            elseAdd $path_j$(v,v') to Edge(v,v')
            end if
        end if
    end while
    Obtain adjv, the set of adjacent nodes of node v
    for L in adjv do
        if L satisfies the length constraint and L
    satisfies the constraint defined on the node u'
    then
            pathi=$path_j$(v,v')+(v,adjv)
            Add pathi to Q
        end if
    end for
    return Edge(v,v')
```

Algorithm 4. EdgeMatching algorithm.

sequence V_T, the pattern nodes are matched in turn according to the topological sequence of the nodes in the pattern graph.

```
Require: Edge(v,v'),$f_E^P$(u,u'),$f_m^P$
Ensure: $G_{temp}$
    while For each v' in Edge(v,v') that satisfies the
    length constraint, the pathlist pathlist(v,v') do
        $totalSum_{DC}$=0
        $totalSum_{CV}$=0
        while For every path P in pathlist(v,v') do
            Calculate aggregate values $A_{DT}$,$A_{DC}$and$A_{CV}$ for
    each attribute
            $totalSum_{DC}$+=$A_{DC}(v, v')$
            $totalSum_{CV}$+=$A_{CV}(v, v')$
        end while
        Calculate the proportion of each value in
    $totalSum_{DC}$ and $totalSum_{CV}$, and use the random
    function to map into the interval [0,1] to obtain
    $f_{DC}$ and $f_{CV}$.
        for i do from 1 to 1000
            Record the number of times mapped to each
    interval in $f_{DC}$ $second_{DC}$;
        end for
        for j do from 1 to 1000
            Record the number of times mapped to each
    interval in $f_{CV}$ $second_{CV}$;
        end for
        $second_{DC}$ and $second_{CV}$ are ranked separately
        The first n second values with large values and
    a crossing point for this path are taken as the
    candidate path pathlist2
        while F door each path P in pathlist2(v,v')
            if then$f_{DT}^m(A_{DT}(V, V')) \geq$ Dtm,$f_{DC}^m(A_{DC}(V, V')) \geq$
    DCm,$f_{CV}^m(A_{CV}(V, V')) \geq$ CVm
                SuitP(v,v')=P
            end if
        end while
        if S thenuitP(v,v') is not null
            add SuitP(v,v') to $G_{temp}$
        end if
    end while
    return $G_{temp}$
```

Algorithm 5. EdgeAttribute algorithm.

The general idea of Algorithm 4 is to start from node v, breadth-first traverse the edges in the data graph, and record each traversal path $path_j(v, v')$ starting from v. When the end point of the path v' matches u', $path_j(v, v')$ is added to the set of matching paths $Edge(v, v')$ satisfying the path length constraint. After obtaining the matching paths of all full conditions, the EdgeAttribute method needs to be called again to review the multiple constraints defined on the pattern edge (u, u'), and the specific steps are shown in Algorithm 5. For a list of paths from v to v' $pathlist(v, v')$, compute the aggregated values A_{DT},A_{DC},A_{CV} on each path, and then perform Monte Carlo filtering on the attributes T_{DC} and T_{CV} on

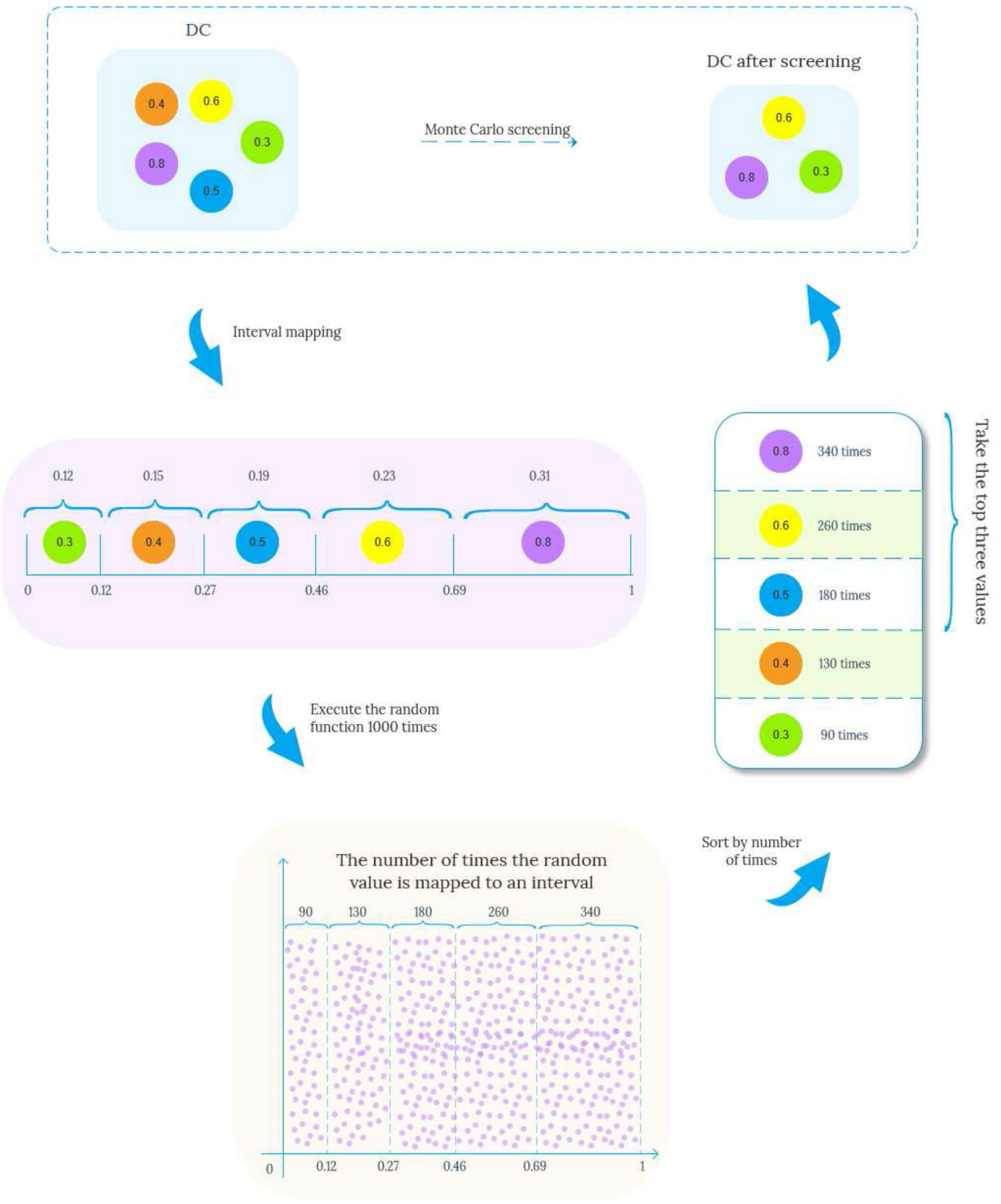

FIGURE 4
The screening process of DC.

the edges. Taking the screening of T_{DC} as an example, the specific process is shown in Figure 4. The proportion value k for each node is obtained by normalizing each attribute value by dividing it by the sum. These values k are then mapped onto an interval f of [0, 1], proportional to its size. Generate multiple random values in the range [0, 1] by repeatedly executing a random function. The generated values are mapped to intervals f and the proportion of values mapped to each interval is calculated, selecting the attribute value corresponding to the top n (adjustable parameter) values that exhibit a significant proportion as the new attribute value, as shown in lines 3-13 of Algorithm 5. Subsequently, the range of constraint values is judged, and the paths that conform to the constraint values are added to G_{temp}.

Example 3.2. For matching paths e1(SLCLIA,Drug), e2(SLCLIA,Operation) and e3(SLCLIA,Drug,Operation) of edge (TC,TM) in pattern Figure 1, where Φ={A_{DT},A_{DC},A_{CV}}, Φ represents the set of aggregated attribute values on the path, Φ_{e_1} = {0.5, 0.7, 0.5}, Φ_{e_2} = {0.5, 0.6, 0.6}, Φ_{e_3} = {0.5, 0.48, 0.54}. Suppose we want to select two candidate edges at random, then we first sum the T_{DC} and T_{CV} of the three edges to obtain $totalSum_{DC}$ = 1.78, $totalSum_{CV}$ = 1.64, Then the proportion of T_{DC} and T_{CV} of the three edges in $totalSum_{DC}$ and $totalSum_{CV}$ is P_{e1}=(0.39, 0.3), P_{e2}=(0.34, 0.37), P_{e3}=(0.27, 0.3), respectively. Similarly, we can get the mapping interval f_{DC} = [e1 (0-0.39), e2 (0.4-0.73), e3 (0.74-1)] and CV mapping interval f_{CV} = [e1 (0-0.3), e2 (0.31-0.67), e3 (0.68-1)]. Executing the random function from 0 to 1,000 times yields the number of times it maps to the two interval values $second_{DC}$=(431, 330,239), $second_{CV}$=(378,511,111), assuming the first two largest numbers. Then path e1 and path e2 enter the next step as candidate paths for normal edge condition matching to filter out edges with too low constraint value.

3.2 Description of the THM algorithm

THM algorithm is a combination of LcKG-McGPM algorithm and TEM algorithm, and Monte Carlo method is applied in node matching and edge matching. Therefore, the main difference between the THM algorithm and the TEM algorithm lies in the screening of candidate nodes, that is, the difference in the GetNodeCandidate function. The rest of the part is not discussed here, please refer to Algorithm 6.

```
Require: Node whose mode is to be matched u∈ $V_P$, Node set V of the data graph
Ensure: The set of candidate nodes $Cand_u$ of u
  while For each node V in v, if V.ited =false do
    if $label_V(u) \in label_V(v)$ then
      Add v to u's set of candidate nodes, nodeCandidate
    end if
  end while
  Example Initialize totalSum=0
  for $u_i$ in nodeCandidate do
    totalSum+=$u_i$.factor
  end for
  The proportion of each value is calculated and mapped to the interval [0,1] using the random function to get ratioValues
  for i from 1 to 1,000 do
    Records the number of times mapped to each value in ratioValues (second)
  end for
  Sort the second
  Add the first n second largest nodeCandidate to $Cand_u$
  return $Cand_u$
```

Algorithm 6. GetNodeCandidate algorithm.

4 Experiments

In this section, our experiments were conducted using a PC running Windows 10, equipped with an Intel Core i9-10900F CPU clocked at 2.80 GHz and 32 GB of RAM. In order to ensure the authenticity and fairness of the experimental data, the data sets used in this paper are LCKGdataset1, LCKGdataset2 and LCKGdataset3 created by Li et al. (2024). Each dataset possesses a unique edge and node configuration with detailed specifications shown in Table 1. To reduce potential errors and ensure reliability, the reported results are arithmetic averages from ten iterations of each graph with different configurations.

TABLE 1 Data sets adopted in experiments.

Data set	Number of nodes	Number of edges	Descriptive information
LCKGdataset1	6,020	178,414	Tumor knowledge graph data set 1
LCKGdataset2	75,000	934,577	Tumor knowledge graph data set 2
Slashdot	77,360	905,468	A friend/foe social network
LCKGdataset3	78,000	949,289	Tumor knowledge graph data set 3

To compare the time efficiency of various algorithms on different datasets, we calculated the duration required for each algorithm to return an equal number of matched subgraphs, and the results are shown in Figure 5. Here, the vertical axis represents the execution time of the algorithm (in seconds), while the horizontal axis represents the number of matched subgraphs (NUM) , and from (a) to (d) represent the results of running on the datasets LCKGdataset1, LCKGdataset2, Slashdot, and LCKGdataset3, respectively. From Figure 6, it is obvious that no matter how much NUM is controlled, the time required by our proposed TEM algorithm is significantly less than that of the TKG-McGPM algorithm, that is, the time efficiency of the TEM algorithm is better than that of the TKG-McGPM algorithm.

In the complex process of graph pattern matching, a key factor lies in the characteristic of graph structure itself, that is, the number of edges is usually smaller than the number of nodes. Especially in the four datasets selected in this experiment (LCKGdataset1, LCKGdataset2, Slashdot and LCKGdataset3), this phenomenon of quantitative difference is particularly prominent. For example, in the process of running on LCKGdataset1, after detailed statistics and analysis, the number of nodes and edges involved in matching reaches thousands, while the number of edges is only about half of the number of nodes. The same is true for several other datasets, which means that graph pattern matching is theoretically less computationally expensive for edges than for nodes.

Our TEM algorithm focuses on the Monte Carlo method applied to the edge matching process, while the TKG-McGPM algorithm uses relevant mechanisms and involves more computational considerations in the node matching process. Since Monte Carlo method itself is based on random sampling to operate, each operation will incur a corresponding computational cost. When dealing with nodes, due to the large number of nodes, more elements need to be traversed and analyzed, which leads

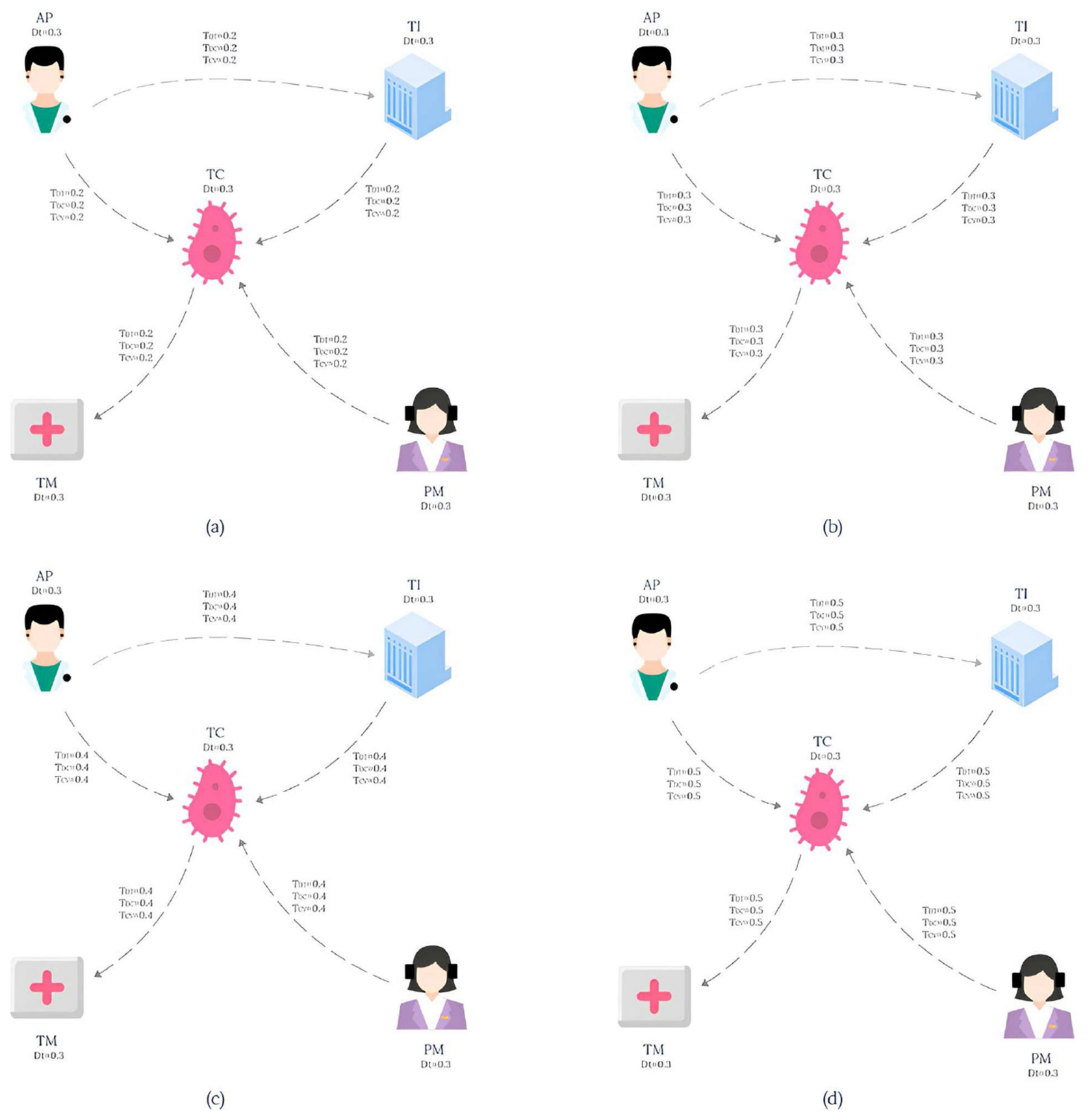

FIGURE 5
Pattern graph for the experiment.

to the accumulation of computing time. As for edges, due to the relatively small number of edges, the number of random sampling, condition judgment and corresponding calculation operations will be reduced when using the Monte Carlo method, so the calculation time required by the Monte Carlo method for edges will be less than that for nodes.

Based on the above analysis, in the actual matching process, TEM algorithm can effectively avoid a large amount of computing consumption caused by too many nodes by virtue of its advantage of using Monte Carlo method in edge matching, so as to produce less time overhead. This reduction in time cost is not a small, local change, but plays a positive role in the entire matching process. From the overall point of view, this makes the whole matching process run more efficiently, and then improves the overall matching efficiency. It not only means that the results can be output faster in a single matching task, but also in the face of large-scale graph pattern matching requirements or scenarios that require multiple repeated matching operations, the time cost saved by TEM algorithm will be more considerable, which provides a more timely and feasible solution for practical applications.

In this study, we set the number of matching results (NUM) to 100, 200, 300, and 400, respectively, and randomly draw 50 matching edges from each result set. To analyze the matching results more systematically, we conducted an in-depth analysis of the mean distribution of T_{DC} values and T_{CV} values, and this statistic is defined as MeanVal. As shown in Figure 7, the X-axis represents the number of matching results (NUM), and the Y-axis represents the mean MeanVal. Compared with the TKG-McGPM algorithm, our proposed TEM algorithm shows a wider distribution

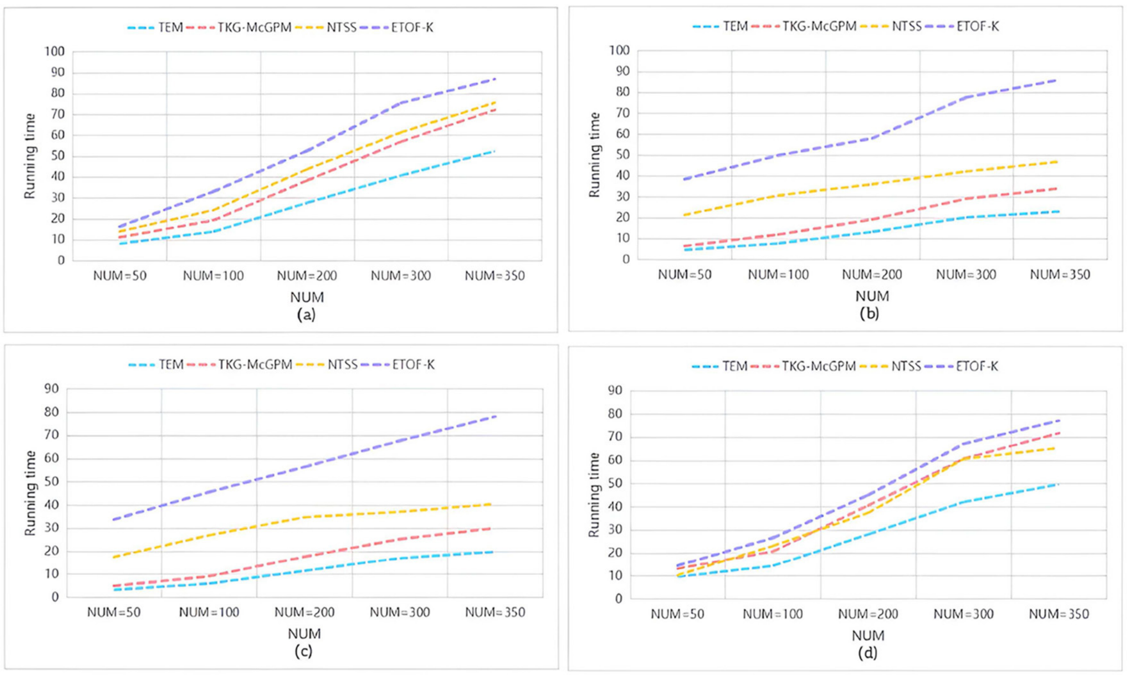

FIGURE 6
The change in the time taken to return different numbers of matching subgraphs on different data sets.

range, and most of the values are concentrated in the interval with high value ratios from 0.7 to 1.0.

Firstly, this shows that the key element of the diversity of matching results is fully taken into account in the design and operation of the TEM algorithm. In the actual application scenarios of graph pattern matching, the diversity of matching results means that users can be provided with more potential matching options with different perspectives and different characteristics, which is particularly important for complex application fields such as medical knowledge graphs. For example, in medical decision AIDS, diverse matching results can cover multi-dimensional information such as different treatment plans and different diagnostic ideas, so as to help medical personnel make more comprehensive considerations. The reason why TEM algorithm can achieve such diversity is that its internal mechanism is not limited to a single matching path or simple matching rule in the matching process, but through reasonable strategies, it can mine and present different possibilities from multiple levels when generating matching results, avoiding the homogenization of results. Therefore, the distribution range of the final MeanVal is broadened to cover a wider range of numerical intervals, especially in the high-value ratio interval, which shows a relatively concentrated trend, which also reflects that the algorithm can effectively filter out those matching edges with low relative value and focus on the matching results with more value and higher quality. The matching edge with high value can be highlighted in the result, and the reference value of the overall matching result is improved.

In contrast, NTSS and ETOF-K can also achieve high mean values in some cases and seem to provide good quality matching results on the surface, but a closer look at the distribution of their MeanVal shows a narrow distribution. This phenomenon is not accidental, and it profoundly reflects the characteristics of the strategies adopted by these algorithms in optimizing the matching results and the possible limitations. In the process of optimizing matching, these two algorithms focus more on the direct improvement of the quality of matching results, for example, they may try to make each output matching result as close as possible to a preset high quality standard by means of stricter screening rules and more accurate matching conditions. However, this approach to some extent makes them fall short in mining the diversity of potential matches. Because the algorithm focuses too much on the optimization of quality, it is easy to follow a relatively fixed and limited path to find results when searching the matching space, thus ignoring other potential matches that may exist, although they are not optimal in some indicators, but have unique value. Finally, the matching results show the characteristics of concentration and narrow range in the numerical distribution. It can not cover more possibilities as comprehensively as TEM algorithm, so it may provide relatively limited reference information in the face of complex and changing practical application requirements.

Further analyzing the root cause of this difference, for the TKG-McGPM algorithm, diversity is given higher importance in the node matching stage, which is worth recognizing, because as the key elements in the graph structure, the diverse matching combinations of nodes can bring rich changes to the whole matching results. However, the algorithm has some shortcomings in the edge matching link, which only considers the size of the comprehensive value of diagnosis and treatment. In the actual graph pattern matching, the edge not only carries simple numerical

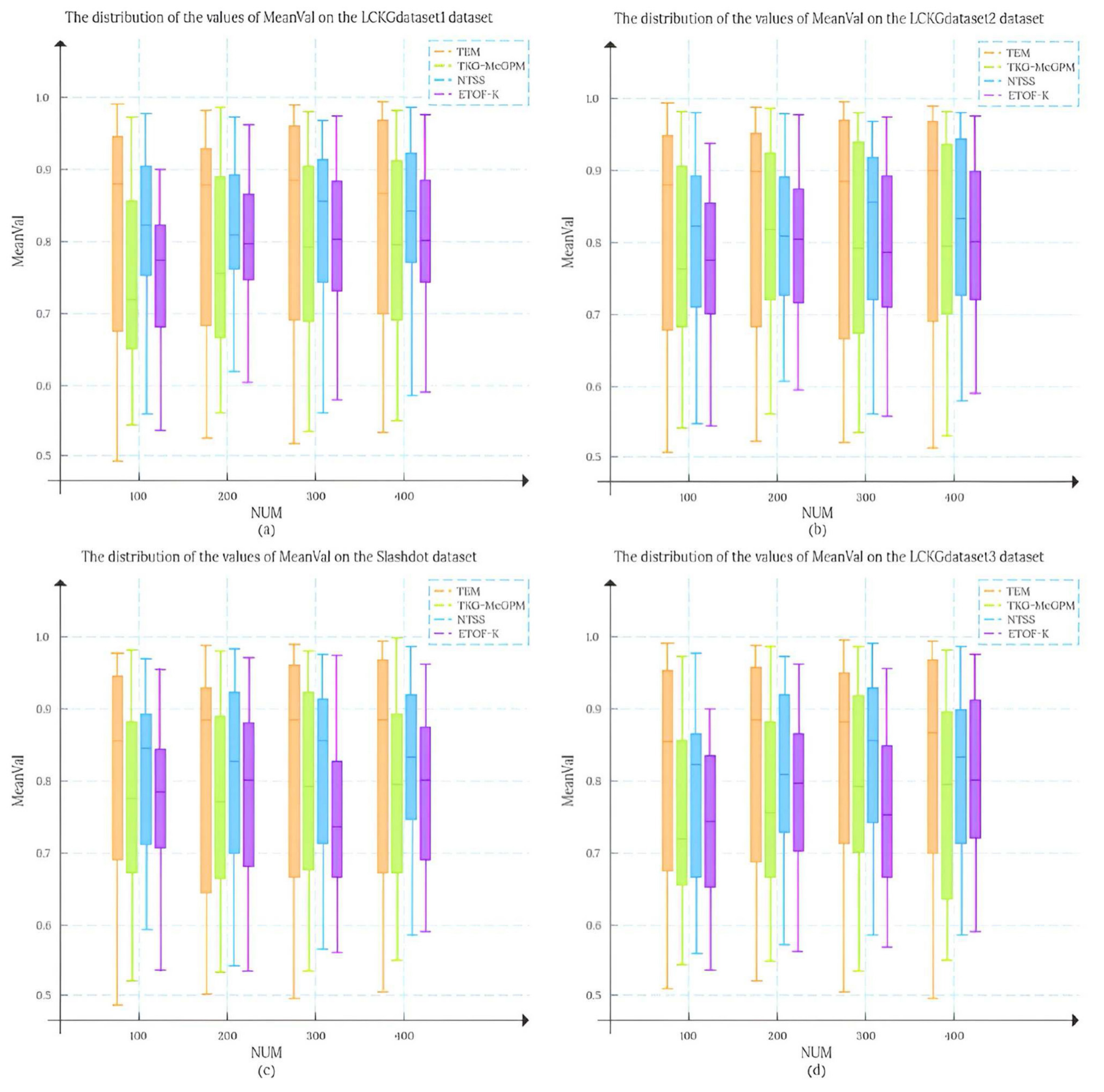

FIGURE 7
The distribution of MeanVal values on different data sets.

information, but also the connection relationship between the edge and the node, the topological structure position and other factors affect the diversity and rationality of the matching. However, the TKG-McGPM algorithm does not fully consider the influence of these edge-related multi-dimensional features on the matching diversity, which leads to the loss of many potential possibilities that can enrich the matching results in the process of edge matching, and then the overall matching results are limited in diversity.

In Monte Carlo simulation, randomness is a key parameter factor that affects the performance of the algorithm. Immediately following, we analyze the effect of different random execution times on the distribution of constraint values on edges , as shown in Figures 8–11. The X-axis of each plot is the mean, MeanVal, and the Y-axis is the number of matched subplots.In Monte Carlo simulations, the range involved in random execution counts is determined by multiple experiments. Specifically, when the value of random execution count is too small, its impact on the final result is relatively weak. If the count is too large, it will undoubtedly cost more time. Based on several rigorous experimental explorations and comprehensive consideration of various factors, we carefully selected four relatively ideal counting results of 1,000, 2,000, 3,000, and 4,000 to carry out comparative experiments. Through such an experimental setting, the influence of random execution times on constraint values can be more intuitively presented, which is helpful

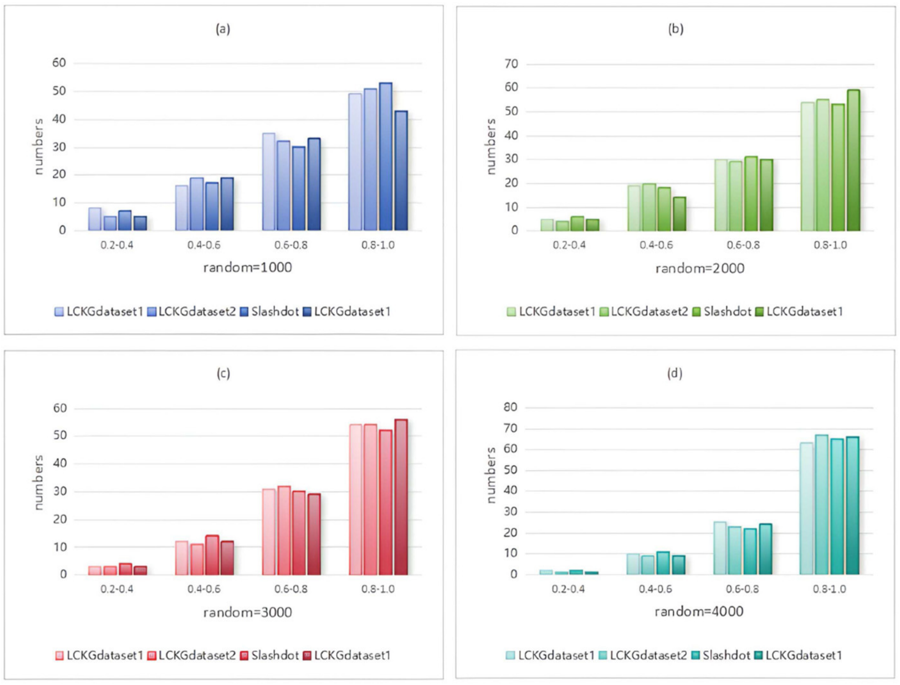

FIGURE 8
When NUM=100, different random executions on different datasets affect the distribution of constraint values on edges.

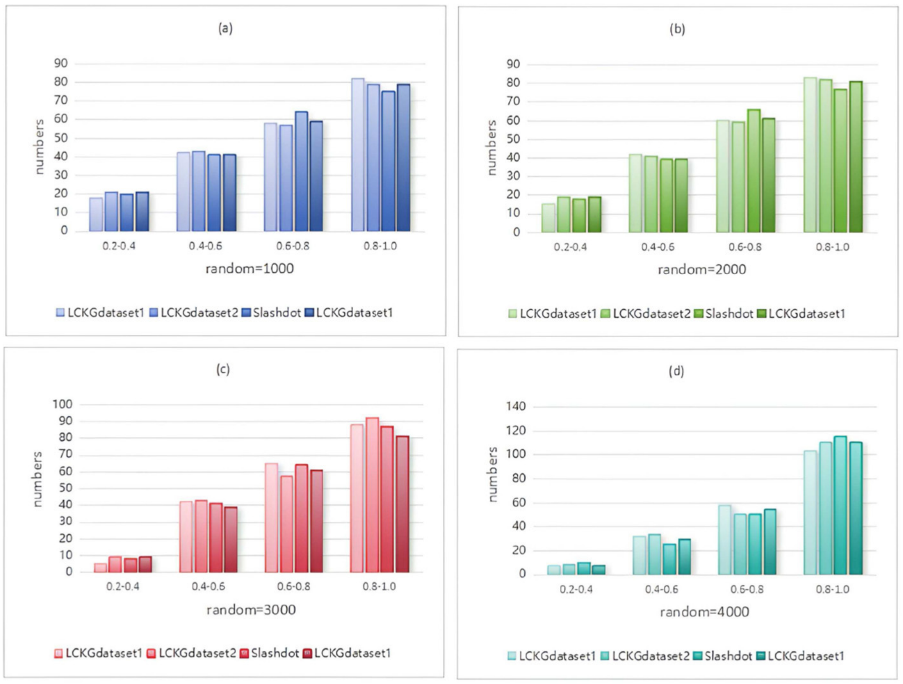

FIGURE 9
When NUM=200, different random executions on different datasets affect the distribution of constraint values on edges.

for us to analyze the internal relationship between relevant variables deeply and accurately, so as to provide a strong basis for subsequent research and conclusion derivation. It can be seen from the figure that when controlling the number of matching results (NUM), the value of MeanVal tends to stabilize in a higher numerical interval as the number of random executions increases. This indicates that

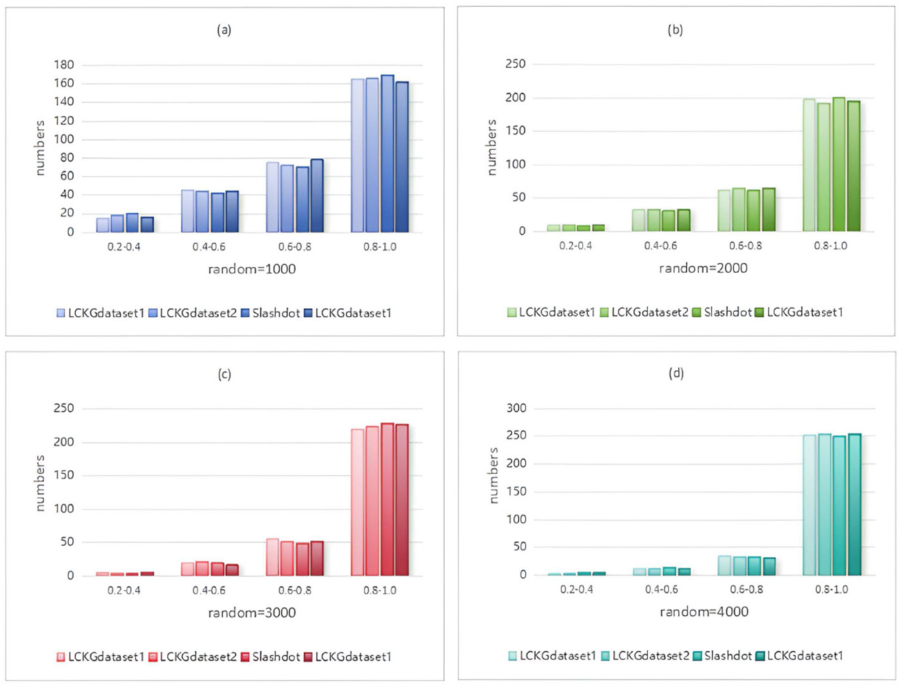

FIGURE 10
When NUM=300, different random executions on different datasets affect the distribution of constraint values on edges.

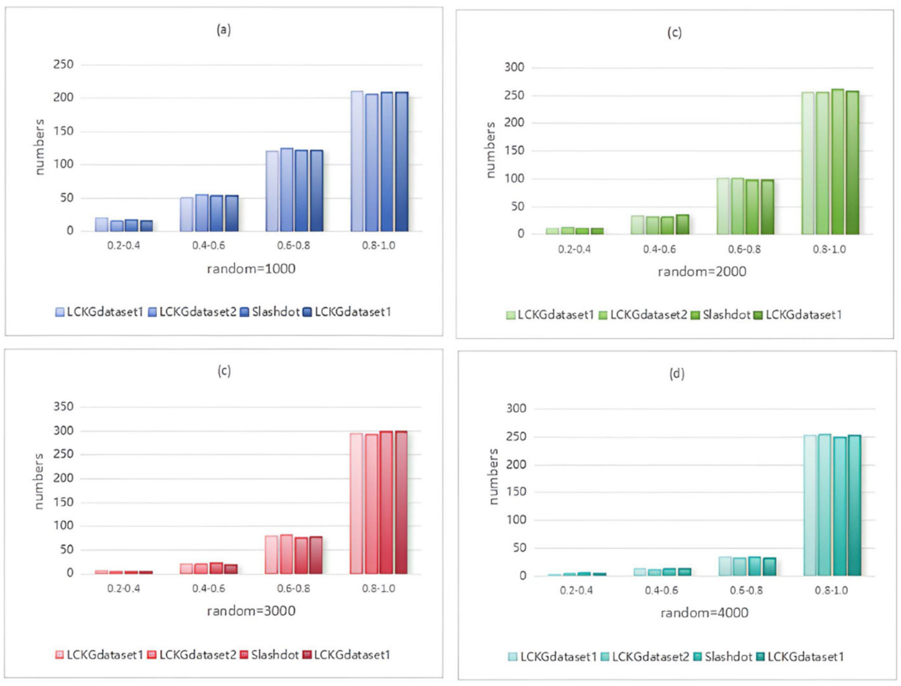

FIGURE 11
When NUM=400, different random executions on different datasets affect the distribution of constraint values on edges.

the increase of the number of random executions can improve the accuracy of the matching results, that is, the more random times, the closer the results are to the optimal solution. Therefore, we can satisfy patient preferences by providing different random number options. For patients with high matching level priority, more random samples can be selected to obtain matching results

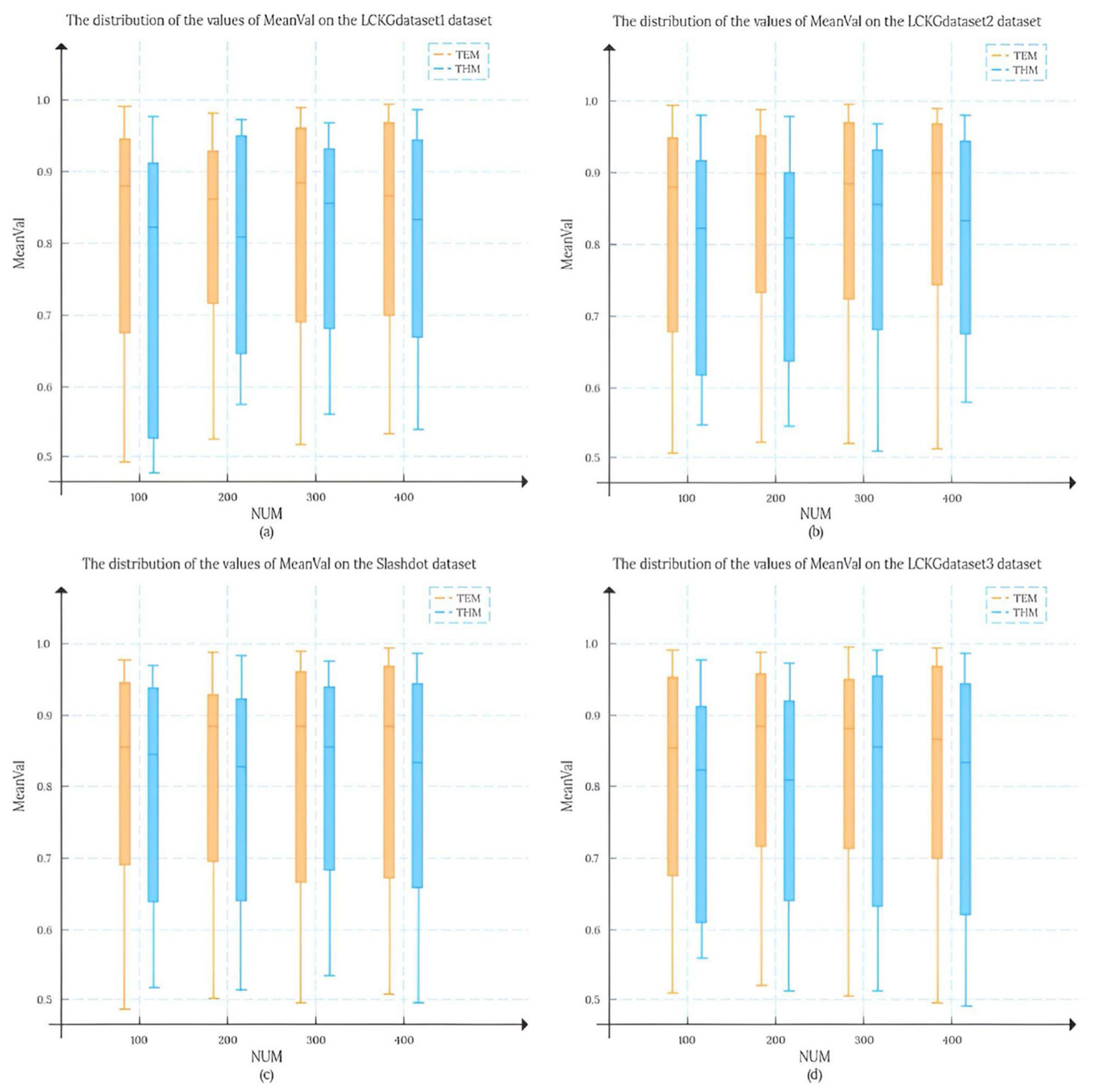

FIGURE 12
The distribution of MeanVal values between the two algorithms on different datasets.

with higher T_{DC} and T_{CV} values. However, for patients who do not care much about the matching values on the edges, a lower random frequency can be chosen to ensure a more uniform distribution of T_{DC} and T_{CV} in the matching results.

From a macro point of view, this method of flexibly adjusting the number of random executions according to different patient preferences not only fully demonstrates the flexibility and adaptability of the matching algorithm, but also has important practical significance in practical applications. Its flexibility is reflected in the ability to quickly adjust the running parameters of the algorithm to generate matching results that meet specific requirements. Adaptability shows that the algorithm can be widely applied to different patient groups with different preference types. Whether patients focus on high-quality matching or balanced and stable matching results, the algorithm can effectively meet the needs. At the same time, by allowing patients to understand the credibility of the matching results more clearly and comprehensively, it further enhances the patient's acceptance and sense of identity for the whole matching process and the final result, which helps to improve the practicability and promotion value of the algorithm in the actual medical scene, so that it can better serve the patient groups with different needs. It

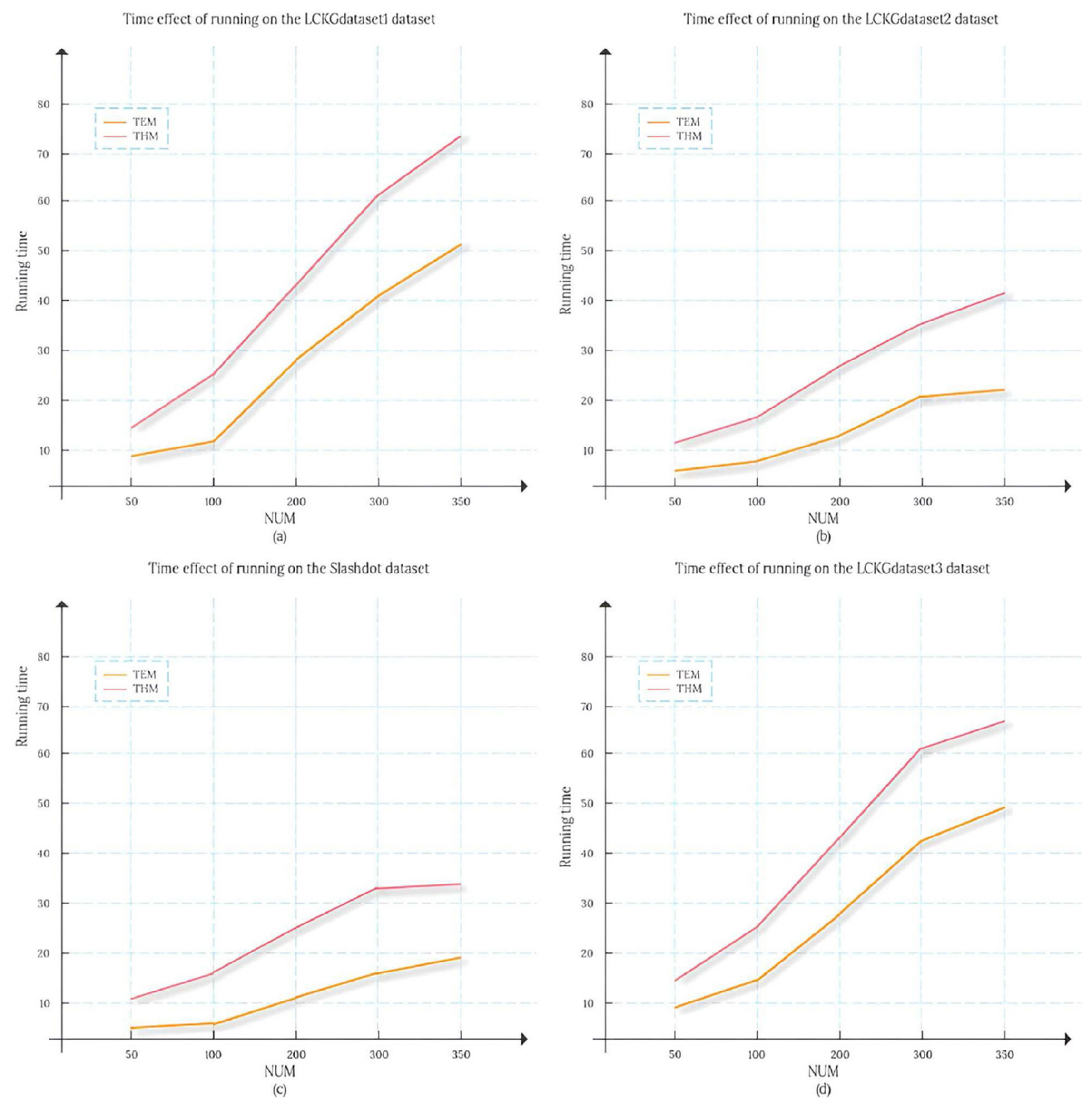

FIGURE 13
Temporal effects of running on different datasets.

provides strong support for medical decision-making and other related applications.

Considering the potential application of Monte Carlo method in node and edge matching, we combine the TKG-McGPM algorithm with the TEM algorithm, and propose the THM algorithm by using Monte Carlo method on both edges and nodes. In order to comprehensively and objectively evaluate the actual performance of THM algorithm, we carefully design and carry out a series of rigorous performance analysis work, focusing on the comparative analysis with TEM algorithm.

The reason why TEM algorithm is selected as the comparison object is that TEM algorithm has shown certain advantages and characteristics in previous research and practical application. Through the comparison of the two, the characteristics, advantages and disadvantages of THM algorithm can be more clearly highlighted. Specifically, the THM algorithm is able to generate more diverse solutions by applying Monte Carlo methods on the nodes and edges, which is clearly demonstrated in Figure 12. We can see that compared with TEM algorithm, the matching results generated by THM algorithm show more abundant changes in the interval distribution of the mean MeanVal. However, any algorithmic feature always comes with a certain cost. For THM algorithm, its advantage of producing diverse solutions is at the cost of consuming a large amount of computing resources. For THM algorithm, its advantage of producing diverse solutions comes at the cost of consuming a lot of computing resources. The reason behind this is that the Monte Carlo method is essentially a random sampling-based method, and its working principle dictates that a large number of simulation operations are required in order to obtain reliable and statistically significant results. Each simulation sampling process involves the calculation and judgment of the attributes and relationships of nodes and edges. With the continuous accumulation of simulation times, the consumption of computing resources increases geometrically. Especially in the face of large-scale medical knowledge graphs or complex graph structures, the consumption of such computing resources will be more significant. For example, when dealing with the actual medical knowledge graph containing massive medical data nodes and complex edge relationships, the THM algorithm may need to occupy a large amount of memory space for storing intermediate calculation results. At the same time, the processor also needs to be in a high load state for a long time to complete numerous simulation sampling and result statistics.It is this high dependence on computing resources and large consumption that directly leads to the obvious disadvantage of THM algorithm in terms of time efficiency, which is clearly presented in Figure 13.

Considering the above factors such as time efficiency and practicality of results, although THM algorithm has a non-negligible advantage in diversity, TEM algorithm has undoubtedly become a more ideal choice due to its excellent performance in these key dimensions. It can better balance the relationship between algorithm performance and practical application requirements, provide efficient and practical solutions for graph pattern matching tasks in medical knowledge graph and other related fields, and better meet the current requirements for fast response and accurate decision-making in practical applications of algorithms.

5 Conclusions

In the context of current research on graph pattern matching algorithms, this study takes the existing KG-McGPM algorithm as the cornerstone, and through in-depth analysis and innovative exploration, proposes two improved graph pattern matching algorithms: TEM and THM, which are mainly used in subgraph matching of lung cancer knowledge graph. By introducing the DC and CV parameters on the edge of the Monte Carlo method, the diversity of matching results is increased. In order to improve the matching effect, the original graph pattern matching model has been modified, and a marginal graph pattern matching algorithm based on lung cancer knowledge graph (TEM) has been proposed. To further verify the effectiveness and efficiency of TEM, a hologram pattern matching algorithm (THM) has been proposed, and the Monte Carlo method has been applied to nodes and edges. Experimental results show that the performance of the TEM algorithm is better than that of the existing algorithms and the THM algorithm. Although the algorithm proposed in this study has achieved excellent performance in the specific field of lung cancer knowledge graph, there is still room for further exploration and improvement from the perspective of macro academic research and practical application expansion. Future research work can focus on extending these algorithms to other medical fields and different types of knowledge graph application scenarios, and deeply investigate their versatility and adaptability through practical verification in diverse contexts. This not only helps to further verify the scientific and reliability of the algorithm itself, but also provides an effective solution for graph pattern matching problems in more fields.

In addition, with the continuous development of computer technology and the increasing scale of data, the calculation speed and matching quality of the algorithm are always the key aspects that need to be paid attention to. Future research can focus on adopting more efficient data structures, such as exploring new graph storage structures or index methods, to optimize the efficiency of the algorithm in data storage and reading. At the same time, combined with advanced optimization algorithms, such as greedy algorithm, dynamic programming algorithm and other ideas, the calculation process of the existing algorithm is deeply optimized, and the overall calculation speed is effectively improved.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

HT: Writing – original draft. LL: Writing – original draft. ZT: Writing – review & editing. ZZ: Writing – review & editing.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This work has been supported by the National Natural Science Foundation of China under grant 62120106008, the Scientific Research Project of Anhui Provincial Health Commission under grants AHBWJ2022b058 and AHWJ2023A10102, and the Joint Fund for Medical Artificial Intelligence of the First Affiliated Hospital of USTC under grant MAI2022Q009.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The author(s) declare that no Gen AI was used in the creation of this manuscript.

Publisher's note

References

Carletti, V., Foggia, P., Saggese, A., and Vento, M. (2018). Challenging the time complexity of exact subgraph isomorphism for huge and dense graphs with VF3. *IEEE Trans. Pattern Analy. Mach. Intellig.* 9, 804–818.

Cheng, J., Yu, J. X., Ding, B., Yu, P. S., and Wang, H. (2008). "Fast graph pattern matching," in *IEEE 24th International Conference on Data Engineering* (Cancun: IEEE), 913–922.

Chikhaoui, B., Tshimula, J. M., and Wang, S. (2020). "Community mining and cross-community discovery in online social networks," in *International Conference on Network-Based Information* Systems, 8. Available at: https://api.semanticscholar.org/CorpusID:221218843 (accessed January 30, 2025).

Fan, W., Li, J., Ma, S., Tang, N., Wu, Y., Wu, Y., et al. (2010). Graph pattern matching: from intractable to polynomial time. *Proc. VLDB Endow.* 3, 264–275. doi: 10.14778/1920841.1920878

Fan, W., Wang, X., and Wu, Y. (2013). Incremental graph pattern matching. *ACM Trans. Database Syst.* 38:47. doi: 10.1145/2508020.2489791

Foggia, P., Sansone, C., and Vento, M. (2001). "An improved algorithm for matching large graphs," in *Proc of the 3rd IAPR-TC-15 International Workshop on Graph-based Representation, Ischia, Italy* (Ischia: Springer), 1–8.

Guo, Z. (2024). *Research on Quality Prediction of Continuous Casting Billet Based on Graph Pattern Matching.* Beijing: *Metallurgical Automation Research and Design Institute*.

Henzinger, M. R., Henzinger, T. A., and Kopke, P. W. (1995). "Computing simulations on finite and infinite graphs," in *IEEE 36th Annual Foundations of Computer Science* (Milwaukee, WI: IEEE), 453–462. doi: 10.1109/SFCS.1995.492576

Hu, J. and Ferguson, A. L. (2016). Global graph matching using diffusion maps. *Data Analy.* 20, 637–654. doi: 10.3233/IDA-160824

Jin, H., Huo, H., and Fang, T. (2023). Strong simulation matching of temporal pattern graph with temporal priority constraints. *Comp. Technol. Dev.* 06, 88–94.

Khan, A., Golab, L., Kargar, M., Szlichta, J., and Zihayat, M. (2020). Compact group discovery in attributed graphs and social networks. *Inform. Proc. Managem.* 57, 0306–4573. doi: 10.1016/j.ipm.2019.102054

Li, L., He, J., Wang, M., and Wu, X. (2016). Trust agent-based behavior induction in social networks. *IEEE Intellig. Syst.* 31, 24–30. doi: 10.1109/MIS.2016.6

Li, L., Tu, H., Tao, Z., Bu, C., and Wu, X. (2024). "Probabilistic graph pattern matching via tumor knowledge graph," in *ACM Transactions on Probabilistic Machine Learning* (New York: ACM).

Liu, F., Xue, S., Wu, J., Zhou, C., Hu, W., Paris, C., et al. (2021). "Deep learning for community detection: progress, challenges and opportunities," in *Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI'20)* (Yokohama: IJCAI), 4981–4987. doi: 10.24963/ijcai.2020/693

Liu, G., Li, L., Liu, G., and Wu, X. (2022). Social group query based on multi-fuzzy-constrained strong simulation. *ACM Trans. Knowl. Discov. Data* 16:27. doi: 10.1145/3481640

Liu, G., Li, L., and Wu, X. (2020). Multi-fuzzy-constrained graph pattern matching with big graph data. *Intell. Data Anal.* 24, 941–958. doi: 10.3233/IDA-194653

Liu, G., Zheng, K., Wang, Y., Orgun, M. A., Liu, A., Zhao, L., et al. (2015). "Multi-Constrained graph pattern matching in large-scale contextual social graphs," in *IEEE 31st International Conference on Data Engineering, Seoul, Korea* (Seoul: IEEE), 351–362.

Sato, R., Habe, H., Mitsugami, I., Satake, S., Sumi, K. and Yagi, Y. (2016). Social group discovery extracting useful features using multiple instance learning. *J. Japan Soc. Fuzzy Theory Intellig. Inform.* 28, 920–931. doi: 10.3156/jsoft.28.920

Su, X., Xue, S., Liu, F., Wu, J., Yang, J., Zhou, C., et al. (2022). A comprehensive survey on community detection with deep learning. *IEEE Trans. Neural Netw. Learning Syst.* 09, 2162–2388. doi: 10.1109/TNNLS.2021.3137396

Tian, Y. and Patel, J. M. (2008). "A tool for approximate large graph matching," in *2008 IEEE 24th International Conference on Data Engineering* (Cancun: IEEE), 963–972.

Tong, H., Faloutsos, C., Gallagher, B., and Rad, T. E. (2007). "Fast best-effort pattern matching in large attributed graphs," in *13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '07)*, 737–746. doi: 10.1145/1281192.1281271

Wei, H., Yu, J. X., Lu, C., and Jin, R. (2014). Reachability querying: an independent permutation labeling approach. *Proc. VLDB Endow.* 7, 1191–1202. doi: 10.14778/2732977.2732992

Yan, M. (2023). *Research on Multi-Constrained Graph Pattern Matching for Large Graph Data* (MA thesis). Hefei University of Technology, Hefei, China.