Approximate Maximum Common Sub-graph Isomorphism Based on Discrete-Time Quantum Walk

Kai Lu*, Yi Zhang*, Kai Xu*, Yinghui Gao† and Richard Wilson‡
*Science and Technology on Parallel and Distributed Processing Laboratory and College of Computer
National University of Defense Technology, Changsha, P.R. China
Email: zhangyinudt@nudt.edu.cn
†College of Electronic Science and Engineering
National University of Defense Technology, Changsha, P.R. China
Email: yhgao@nudt.edu.cn
‡Department of Computer Science
University of York, York, UK, YO10 5GH
Email: richard.wilson@york.ac.uk

***Abstract*—Maximum common sub-graph isomorphism (MCS) is a famous NP-hard problem in graph processing. The problem has found application in many areas where the similarity of graphs is important, for example in scene matching, video indexing, chemical similarity and shape analysis. In this paper, a novel algorithm Qwalk is proposed for approximate MCS, utilizing the discrete-time quantum walk. Based on the new observation that isomorphic neighborhood group matches can be detected quickly and conveniently by the destructive interference of a quantum walk, the new algorithm locates an approximate solution via merging neighborhood groups. Experiments show that Qwalk has better accuracy, universality and robustness compared with the state-of-the-art approximate MCS methods. Meanwhile, Qwalk is a general algorithm to solve the MCS problem approximately while having modest time complexity.**

***Keywords*-discrete-time quantum walk; maximum common sub-graph; quantum interference;**

I. Introduction

In the real world, graphs or networks are widely used to represent structural and relational information in a way that is abstracted from the actual data. The information, such as social networks, molecular structures, transport networks or protein-protein iterations, can be stored and processed more naturally in this form. The maximum common sub-graph (MCS) is a classical NP-hard problem in graph analysis which aims to find the largest common substructure between two graphs [1,2]. Being strongly relevant to graph similarity, MCS is the core process of many real applications which require a measure of similarity between two networks. For example, MCS algorithms have been used for scene matching [3], video indexing and chemical similarity.

The MCS problem has been widely studied over the last 4 decades, in two broad categories, namely exact MCS methods and approximate methods. Prune and backtracking methods were used in many of the exact MCS approaches, such as VF2 [4] and McGregor [5]. Another common approach is to find the maximal clique in the association graph in order to solve the general MCS problems [6]. Although these algorithms can solve this problem generally and exactly, all of them have exponential cost. When the graph size is large, these algorithms are unavailable because of the unacceptable time consumption.

In contrast to exact MCS algorithms, some approximate MCS methods have been proposed, which are more practical for real applications. Polynomial time consumption is achieved by the trade-off of getting some locally optimal solutions rather than the global optimal one. Lu [7] exploited the properties of planar triangulation graphs and designed an approximate MCS algorithm tailored for this kind of graph. Pellilo [8] utilized replicator dynamics to find the approximate maximal clique in association graph which can imply the maximal common sub-graph. However, there exist two problems with state-of-the-art approximate MCS detectors: Firstly, it is difficult to solve this problem approximately for the general graphs. Secondly, though they are not exponential algorithms, in order to reduce the error as much as possible, the procedure of these algorithms is usually very complex, which results in the overwhelming time consumption even on moderately sized graphs.

From the development of quantum computing, especially Shor's integer factoring algorithm and Grover's database searching algorithm, the field has made significant progress over the last few years [9]. The quantum walk was firstly proposed in 1993, and has drawn much interest from then on. Recently, some novel quantum algorithms using quantum walk to detect graph isomorphism have been designed [10,11]. Via utilizing the quantum parallel and quantum interference, these quantum algorithms could achieve better performance and discriminative power than the traditional ones. However, until so far, there are no quantum algorithms to solve the MCS problem.

In this paper, a discrete-time quantum walk is utilized to propose a novel quantum algorithm Qwalk for the first time, which could solve the MCS problem approximately. The new algorithm has two steps. Firstly, for every vertex pair in the two graphs, the maximum isomorphic neighborhood group match will be detected by using the destructive interference of quantum walk. Secondly, the maximum isomorphic neigh-

borhood group matches will be sorted by size and merged in descending order until no more can be merged into the solution. From the initial detection and merging process, a large common sub-graph should be obtained. Because all the vertex states are stored in the quantum bits, they could be transferred in parallel during a discrete-time quantum walk. Therefore the time complexity of the new quantum algorithm is only $O(NE^{1.5})$ for two graphs with N vertices and E edges individually. Through experiments, it is reported that the new algorithm shows better accuracy, universality and robustness for noise compared with the state-of-the-art approximate MCS algorithms.

II. Background

We begin by introducing some definitions related to graphs and which are relevant to our algorithm.

A. Problem definition

Definition 1. A graph is a tuple $G = (V, E)$, where V represents all the vertices in the graph with a set of the adjacent relation $E \subseteq V \times V$.

In this paper, we will mainly focus on the unweighted, undirected and connected graphs.

Definition 2. A graph $G' = (V', E')$ is an (induced) sub-graph of graph $G = (V, E)$, if and only if $V' \subseteq V$ and $\forall v_1, v_2 \in V', (v_1, v_2) \in E' \Leftrightarrow (v_1, v_2) \in E$.

Definition 3. A (sub)graph $G = (V, E)$ is isomorphic to $H = (V', E')$, if there exists as least one bijective function $f : V \to V'$ so that $\forall v_1, v_2 \in V, (v_1, v_2) \in E \Leftrightarrow (f_{v_1}, f_{v_2}) \in E'$.

Definition 4. For two graphs G and H, the MCS problem aims to find out the largest sub-graph G' in G which is isomorphic to at least one sub-graph H' in H. Here the size of a graph is measured by the cardinality of the vertex set.

The distance $d(u, v)$ between two vertices on a graph is the length of the shortest path in the graph which joins the two vertices, i.e. the smallest number of edges traversed to move from one vertex to the other.

Definition 5. The 1-level neighborhood group of the vertex in a graph G is a sub-graph $G_v^1(V', E')$ with $V' = \{v_i | \forall v_i \in V, d(v, v_i) \leq 1\}$. Similarly, the k-level neighborhood group is $G_v^k(V', E')$ and $V' = \{v_i | \forall v_i \in V, d(v, v_i) \leq k\}$.

Definition 6. An maximal isomorphic neighborhood group match is called an ING and marked as ING(u, v, k), where u and v are the vertices of two graphs G_1 and G_2 while $k = \max\{i | G_u^i$ is isomorphic to $G_v^i\}$.

B. Discrete-time quantum walk

The random walk is a classical stochastic process in which a walker, residing on a vertex of the graph, takes a random step along one of the edges to a new vertex. This process has previously been used in graph matching [12]. By computing the number of walks with different lengths in the graphs, the random walk could discriminate between different graphs. As the quantum counterpart of random walk, quantum walks have become a solid field of quantum computing full of exciting open problems for physicists, computer scientists and engineers. More recently, attention has focussed on the quantum walk on a graph [10,13,14] which has different properties to its classical counterpart and may be computationally more powerful.

Analogously with the random walk, the quantum walk has a state space and transfer matrix. Because a quantum walk is reversible, the quantum state includes the position and the walk direction. For each step, the particle will walk towards every possible direction with certain quantum amplitudes simultaneously.

For the quantum walk on a graph $G = (V, E)$, all the states are stored in a quantum superposition $|\varphi\rangle$ as (1),

$$|\varphi\rangle = \sum_{(u,v)\in E} \alpha_{uv} |uv\rangle, \alpha_{uv} \in C, \sum_{(u,v)\in E} {\alpha_{uv}}^2 = 1 \quad (1)$$

where $|uv\rangle$ represents the quantum state from vertex u to v with the quantum amplitude α_{uv}. Using the common Grover diffusion matrices, every state will be transferred as (2),

$$|uv\rangle \to (\frac{2}{d(v)} - 1) |vu\rangle + \frac{2}{d(v)} \sum_{\forall k \in V, k \neq u \& (k,v) \in E} |vk\rangle \quad (2)$$

where $d(v)$ denotes the degree of vertex v.

In contrast to the classical random walk, in the quantum walk every different walk will interfere with each other when they meet because the particle has a quantum amplitude, not a probability. This means that graph symmetry can be detected via the interference of a discrete-time quantum walk[10]. Consider the graph in Fig. 1(a), it is obvious that the two parts besides vertex v are symmetric. Assume the initial state of the whole graph is symmetric as (3),

$$\begin{aligned} |\varphi_0\rangle = & \tfrac{a}{\sqrt{2}}[|u_1u_2\rangle + |u_2u_1\rangle + |u_1u_3\rangle + |u_3u_1\rangle] \\ & + \tfrac{a}{\sqrt{2}}[|w_1w_2\rangle + |w_2w_1\rangle + |w_1w_3\rangle + |w_3w_1\rangle] \end{aligned} \quad (3)$$

After every step of quantum walk, we focus on the quantum amplitudes of vertex v as shown in (4),

$$\begin{aligned} \alpha_{u_1v}(t+1) &= \tfrac{2}{3}\alpha_{u_2u_1}(t) + \tfrac{2}{3}\alpha_{u_3u_1}(t) - \tfrac{1}{3}\alpha_{vu_1}(t) \\ \alpha_{w_1v}(t+1) &= \tfrac{2}{3}\alpha_{w_2w_1}(t) + \tfrac{2}{3}\alpha_{w_3w_1}(t) - \tfrac{1}{3}\alpha_{vw_1}(t) \end{aligned} \quad (4)$$

Via the Hadamard operator for the quantum state, the interference amplitude between u_1 and w_1 is computed as follows,

$$\alpha^-(u_1, w_1) = \frac{1}{\sqrt{2}}[\alpha_{u_1v}(t) - \alpha_{w_1v}(t)] \quad (5)$$

We find that the interference amplitude $\alpha^-(u_1, w_1)$ will be equal to 0 after every step of quantum walk. This property of quantum walk has been utilized to detect graph isomorphism via constructing the auxiliary graph and connecting all the vertex pairs with auxiliary nodes [10]. Destructive interference will happen in every auxiliary node which connects the true vertex mapping.

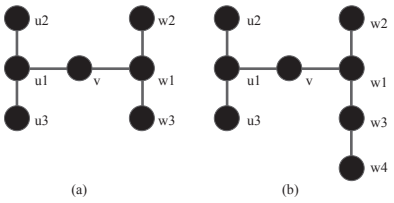

Figure 1. (a)Symmetric graph. (b)Asymmetric graph

III. ALGORITHM

In this section, we find some new observations based on the discussion of MCS problem and the discrete-time quantum walk. Based on them, a novel quantum algorithm Qwalk is proposed to solve the MCS problem approximately.

A. Observations

Observation 1. The quantum amplitude of a vertex will spread to another vertex according to their distance during quantum walk. The discrete-time quantum walk can measure the distance.

From (2), it is known that for every step, the quantum amplitude could be transferred from one vertex v to its 1-level neighborhood group G_v^1. Therefore, if $d(u,v) = k$, the quantum amplitude of vertex v will spread to vertex u after a k-step quantum walk.

Observation 2. The destructive interference of discrete-time quantum walk can be used to detect graph isomorphism, while the k-step discrete-time quantum walk can be used to detect the (k-2)-level isomorphism neighborhood group matches.

In [10], the destructive interference of discrete-time quantum walk on the auxiliary graph will make the interference amplitudes of the auxiliary nodes between the true vertex mappings equal to 0. Consider the quantum walk on the graph in Fig. 1(b), the vertex can be considered as the auxiliary node between the vertex u_1 and w_1. We see that $G_{u_1}^1$ is isomorphic to $G_{w_1}^1$, but $G_{u_1}^2$ is not isomorphic to $G_{w_1}^2$. The different substructure is the vertex w_4 and $d(v, w_4) = 3$. Based on Observation 1, it is known that the quantum amplitude of vertex w_4 will spread to vertex v after 3 steps of discrete-time quantum walk. Because the interference amplitude maintains 0 after two steps, it means all the 2-length walks around the vertex are symmetric and there is no difference between the two substructures $G_{u_1}^1$ and $G_{w_1}^1$. However, the symmetry is destroyed after the 3th step. Therefore, through 3-step discrete-time quantum walk, an isomorphic neighborhood group match $G_{u_1}^1$ and $G_{w_1}^1$ is detected.

Observation 3. Every graph can be constructed from some neighborhood groups. Every common sub-graph pair can be decomposed into some isomorphic neighborhood group matches.

For a connected graph, every vertex can reach all the others via a limited-length path. Therefore, the whole graph can be considered as a neighborhood group of every vertex. In general, we can divide the graph into some neighborhood groups randomly and there are exponential different divisions for a graph. Besides, if a sub-graph is a common one of two graphs, the division of this substructure can also be included into the divisions in the two graphs. Moreover, every element in the neighborhood set in the division is isomorphic in the divisions of the two graphs. That is the reason we can find MCS via merging the isomorphic neighborhood group matches.

B. Method

Given the two graphs G_1 and G_2 with m and n vertices respectively, we need to construct the auxiliary graph G_{12} firstly. We use auxiliary nodes to connect every vertex pair between the two graphs.

The whole algorithm can be divided into two steps.

Step 1. Run the discrete-time quantum walk on the auxiliary graph G_{12}. This step is similar with the algorithm in [10]. However, we set a label L_{uw} on every auxiliary node v_{uw} (v_{uw} connects vertex u and w in G_1 and G_2 respectively) to count the steps the interference amplitude $\alpha^-(u,w)$ maintains 0. Therefore, after the first step of quantum walk, we will check the amplitude of every auxiliary node, if it is equal to 0, its label will increment by 1. Otherwise, the auxiliary node will be marked and not be checked later. Then run quantum walk until all the auxiliary nodes are marked or κ-steps quantum walk has finished ($\kappa = \max(d_1, d_2) + 1$, in which d_1 and d_2 are the diameters of the two graphs respectively).

After the first step of the algorithm, all the auxiliary nodes have labels which denote the times that the interference amplitude maintains 0. From Observation 2, we know that it also means that we have found $m \times n$ isomorphic neighborhood group matches. Every match can be represented as a triple ING$(u, v, L_{uv} - 2)$.

Step 2. After collecting $m \times n$ isomorphic neighborhood group matches as a set Ψ, we will sort them by size. In order to get a larger common sub-graph as a better solution, we give high priority to the bigger INGs to be chosen to merge into the result.

After sorting, we begin to merge the matches in descending order. We represent the isomorphic sub-structure pair from the two graphs as $S = (S_1, S_2)$, this pair is initially empty $S = (\emptyset, \emptyset)$. We build this structure by sequentially merging INGs discovered by the quantum walk.

Firstly, we introduce the core function *merge*(A, B). $A = (A_1, A_2)$ and $B = (B_1, B_2)$ are two isomorphic substructure pairs. If the following 4 conditions are satisfied, the merge will succeed.

(1) A_1 and B_1 are connected, $A_1 \not\subset B_1$ and $B_1 \not\subset A_1$. Same to A_2 and B_2.
(2) Any vertex which appears multiple times in $\{A_1, B_1\}$ maps to the same vertex in $A_2 \cup B_2$.
(3) The vertex number of $A_1 \cup B_1$ should be the same with that of $A_2 \cup B_2$.
(4) Construct the auxiliary graph which only connects the subgraphs $A_1 \cup B_1$ and $A_2 \cup B_2$. Run π-step ($\pi = d_{A_1} + d_{B_1} + 1$) discrete-time quantum walk, all the interference amplitudes of the auxiliary nodes should maintain 0.

1: **procedure** QWALK(G_1, G_2)
2: Construct the auxiliary graph G_{12}
3: $\forall j,k \in [1,N]$, $M_{jk} \leftarrow 0$ and $M'_{j+(k-1)\times N} \leftarrow 0$
4: set $|\varphi_0\rangle$ as a symmetry quantum state on G_{12}
5: **for** $i = 1 : \kappa$ **do** ▷ $\kappa = \max(d_1, d_2) + 1$
6: $|\varphi_{i-1}\rangle \rightarrow |\varphi_i\rangle$ ▷ quantum walk on G_{12}
7: **if** $\forall j,k, M_{jk} \neq 0$ **then**
8: stop quantum walk
9: **end if**
10: **for** every auxiliary node v_{jk} **do**
11: **if** $\alpha^-(j,k) = 0$ and $M_{jk} = 0$ **then**
12: $L_{jk} \leftarrow L_{jk} + 1$
13: update(ING$(j,k,L_{jk}-2)$)
14: **else**$M_{jk} \leftarrow 1$
15: **end if**
16: **end for**
17: **end for**
18: sort(Ψ)
19: set $S = (\emptyset, \emptyset)$ and $i \leftarrow 1$
20: **while** $i \leq N^2$ **do**
21: **if** $M'_i = 0$ and merge$[S, \text{ING}(j,k,L)]$ **then**
22: $S_1 \leftarrow S_1 \cup G_j^L$ and $S_2 \leftarrow S_2 \cup G_k^L$
23: $M'_i \leftarrow 1$ and $i \leftarrow 1$
24: **else**$i \leftarrow i + 1$
25: **end if**
26: **end while**
27: **end procedure**

Figure 2. Qwalk's algorithm. M_{jk} denotes the mark of the auxiliary node v_{jk}. M'_i denotes the mark of the i^{th} triple in the sorted set Ψ.

We then use the merge operation to include the INGs into S. We iterate through all INGs from the largest downwards to find a legal merging step. If *merge*$[S, \text{ING}(u_i, w_i, L_{u_i w_i} - 2)]$ returns true, we set $S = (S_1 \cup G_{u_i}^{L_{u_i w_i}-2}, S_2 \cup G_{w_i}^{L_{u_i w_i}-2})$. Meanwhile we delete ING$(u_i, w_i, L_{u_i w_i} - 2)$ from the set Ψ and restart the search to retry all the remaining . The whole procedure will finish when every remaining INGs cannot be merged into S. S is the final solution of our algorithm. Fig. 2 gives the pseudo code of the whole algorithm.

C. Analysis

We now discuss the time complexity. For simplicity, we assume that both of the two input graphs have N vertices and E edges.

For these graphs, there are totally $2E + 2N^2$ states and the state transition involves matrix multiplication of state-state matrices, the time complexity of quantum walk is about $O\left(E^3\right)$. However, according to [10], it is known that the simulation of the discrete-time quantum walk could be realised with complexity $O(E^{1.5})$ on the auxiliary graph via Grover's technique. Though we need to detect the values of all the interference amplitudes of the auxiliary nodes after every step, compared with transition of quantum state, the time cost can be neglected. Therefore, for Line 5-18, the time complexity of *Step 1* is $O(E^{1.5})$.

The size of the set Ψ is N^2 because there is an potential match between all possible pairs of vertices. The time cost of quick sort is $O(N^2 \log N^2)$. Because there are no more than N times we need to use the 4th condition to judge the merge of two isomorphic substructure match in *Step 2* (According to the 1st condition of the function *merge*, in the worst case, for every merge, we add at least one new vertex mapping into S) and when we have to do it, a run of discrete-time quantum walk is needed, the time complexity for Line 22-29 is no more than $O(NE^{1.5})$. Because $E \leq N^2$, the time consumption of this algorithm is approximately $O(N^4)$.

In [7], Lu's algorithm found the maximum common sub-graph via travelling all the triangulations in the two input graphs. All pairs of ordered triangles need to be considered as the roots of the pairs of traversals. Therefore the algorithm will cost $O(kmn)$ for two graph of size m and n and their maximum common sub-graph of size k. We can consider that the time complexity is about $O(N^3)$.

In [8], in order to obtain the maximum common sub-graph, the association graph is utilized which size is N^2 for two graphs of size N. And such a N^2-length vector will iterate in replicator dynamics function for many times while it requires cubic time for every iteration as reported. In general, more iterations are needed for bigger graphs. Therefore, the time complexity of this algorithm is $O(KN^6)$ and K is the iteration time which cannot be neglected.

Compared with these state-of-the-art approximate MCS algorithms, Qwalk shows good performance relative to other general algorithms.

IV. EXPERIMENTS

In order to evaluate the performance comparison between the new algorithm and the state-of-the-art ones, two common kinds of graphs are tested which are the Delaunay Triangulation Graph (DTG) and the Erdös-Rényi random connected graph (RCG) including some synthetic dataset and COIL-100 database [15].

Because VF2 [4] is a famous exact MCS algorithm, we use its result as the optimal one. Our algorithm (Qwalk) is compared to the algorithm of Lu [7] specialized for DTGs and the maximal clique algorithm of Pelillo [8] (MC-X). Here X refers to the number of replicator function iterations used. The reported error is the fractional size difference between the recovered MCS and the one reported by VF2 as the following equation.

$$Error_A = \frac{abs\left(S_{VF2} - S_A\right)}{S_{VF2}} \tag{6}$$

where S_{VF2} and S_A are the sizes of the maximum common sub-graph matches of VF2 and the approximate algorithm A.

Fig. 3 demonstrates the accuracy comparison of all the approximate MCS algorithms for DTGs with different sizes. Although Lu is a tailored algorithm for DTGs, Qwalk and MC outperform for this kind of graphs. Moreover, Qwalk has a small standard deviation and stable accuracy as the graph size increases. Therefore, the newly proposed algorithm has good accuracy for DTGs.

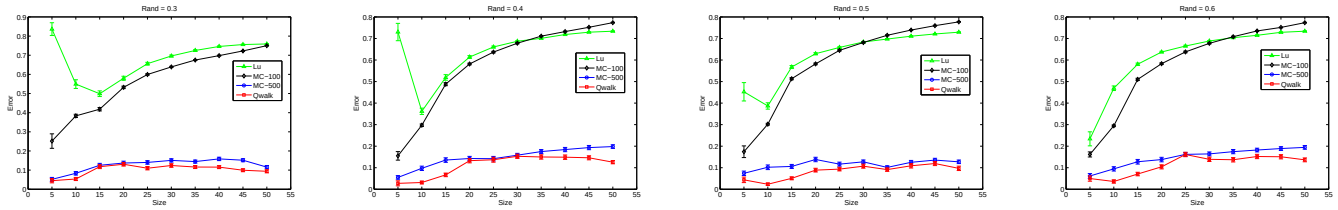

Figure 4. The accuracy comparison for RCGs with connected probability 0.3-0.6.

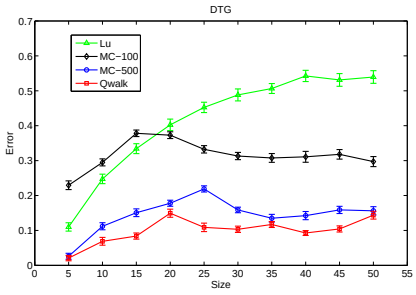

Figure 3. The accuracy comparison for DTGs.

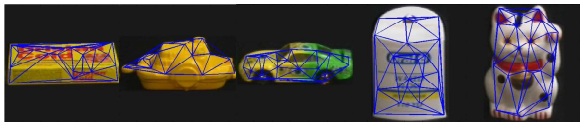

Figure 5. Delaunay triangulation graphs of the objects.

Fig. 4 shows the accuracies of these algorithms for RCGs with different connecting probabilities. Because RCGs do not have the same properties with DTGs, the algorithm Lu will be difficult to find triangulations in the input graphs so that the accuracy is worst and unstable. Meanwhile, it is found that Qwalk performs similar with MC-500. However, Lu and MC with few iterations have bad accuracy and stability. Therefore, Qwalk has better universality.

Maximum common sub-graph between two graphs is strongly relative to the graph similarity. In this part, we will use the solution of these approximate MCS algorithms to be the measurement of graph similarity to making image clustering for COIL-100 database. For each image in this database, the corner features are extracted using the Harris corner detector [16]. The extracted corner features are used as the vertices to construct Delaunay triangulation graphs in our experiment. We choose five objects randomly of which the corner numbers are all around 25, and for each one, five images are taken from -10 degree to 10 degree. Fig. 5 shows the Delaunay triangulation graphs of all the five objects with angulation 0. We use the size ratio between the input graphs and MCS as the similarity and utilize multidimensional scaling (MDS) to make image clustering into 5 groups for every algorithm. The detail distributions of graph are visualized in Fig. 6. We see that only Qwalk could make an approximately correct clustering for these 25 graphs. All the graphs of the same objects are near. However, the results of Lu and MC are bad and the same-object graphs are far away with each other and the different-object graphs are mixed so that it is difficult to make a correct clustering.

In the real world, noise will disturb usually. In order to test the robustness of these algorithms, noise will be added into the DTG database with size 30, 40 and 50. The way to add noise is to delete n edges randomly from DTGs. Fig. 7 reports the accuracy comparison of these algorithms for DTGs with different noise intensities. Because noise will destroy the graph and its original properties, the accuracy of Lu will be abnormal, bad and unstable. While Qwalk and MC performs better. Especially when the graph is more bigger, Qwalk shows better accuracy and stability than MC. Therefore, Qwalk has better robustness for noise.

V. Conclusion

In this paper, a novel quantum algorithm Qwalk is designed which utilizes discrete-time quantum walk firstly to solve the MCS problem approximately. Based on the observation that it can be fast and convenient to detect all the maximum isomorphic neighborhood group matches in parallel via the interference of discrete-time quantum walk, this kind of substructure is used to solve the MCS problem for the first time. For the general graphs, our new algorithm could give an approximate optimal solution in time complexity $O(NE^{1.5})$. Experiments report that compared with the state-of-the-art approximate MCS algorithms, our newly proposed algorithm could achieve better availability for the real world.

Our algorithm will be low-performance if quantum walk could not find some high-level isomorphic neighborhood pairs, especially when most of the node pairs in the two graphs have different degrees. This is a common situation in the problem of sub-graph isomorphism. In the future work, some tricky auxiliary nodes or edges will be utilized to address this problem so that our algorithm will be also proper to the sub-graph isomorphism problem. Besides, we will continue to focus on what will happen if the quantum walk runs on the association graph.

Acknowledgment

The work is supported in part by National High-Tech R&D Program of China (863 Program) under Grants 2012AA01A301 and 2012AA010901. And it is partially supported by National Science Foundation (NSF) China 61103082

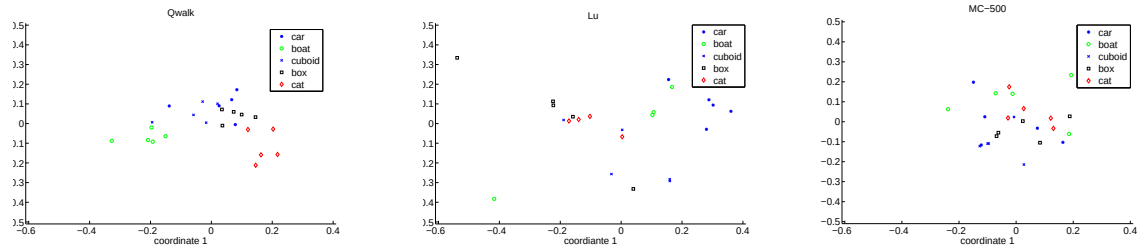

Figure 6. Graph Clustering for five objects in COIL-100 database.

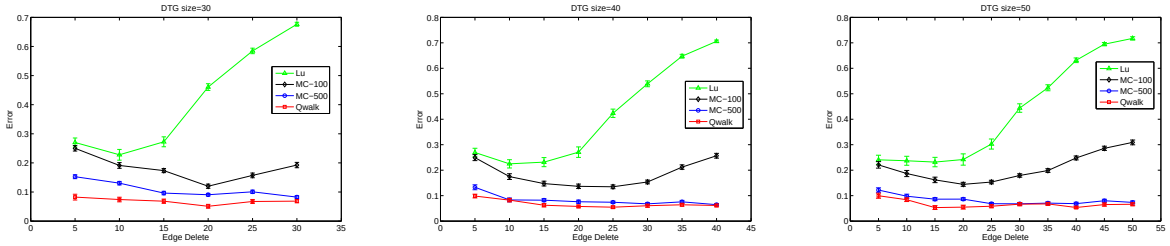

Figure 7. The accuracy comparison for DTGs dataset with edge-delete noise.

and 61170261. It is also a part of Innovation Fund Sponsor Project of Excellent Postgraduate Student (B120601 and CX2012A002).

REFERENCES

[1] D. Conte, P. Foggia, C. Sansone and M. Vento, Thirty years of graph matching in pattern recognition, IJPRAI, Vol. 18 No. 3 (2004) 265-298.

[2] P. Foggia, G. Percannella and M. Vento, Graph matching and learning in Pattern Recognition in the last 10 years, International Journal of Pattern Recognition and Artificial Intelligence, 2013.

[3] R. Horaud and R. Skordas. Stereo correspondence through feature grouping and maximal cliques. IEEE Transactions on Pattern Analysis and Machine Intelligence, 11:1168-1180, 1989.

[4] L. P. Cordella, P. Foggia, C. Sansone, and M. Vento. A (sub)graph isomorphism algorithm for matching large graphs. IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(10): 1367-1372, 2004.

[5] J. McGregor. Backtrack Search Algorithms and the Maximal Common Subgraph Problem. Software Practice and Experience 12(1), 23-34, 1982.

[6] I. Koch. Enumerating all connected maximal common subgraphs in two graphs. Theoretical Computer Science 250(1), 1-30, 2001.

[7] Y. Lu, H. Bunke, and C. Liu. An algorithm for maximum common subgraph of planar triangulation graphs. Graph-Based Representations in Pattern Recognition. Springer Berlin Heidelberg, 162-171, 2013.

[8] M. Pelillo. Replicator equations, maximal cliques, and graph isomorphism. Neural Computation, 11(8), 1933-1955, 1999.

[9] M. Lanzagorta, and J. Uhlmann. Quantum rendering: an introduction to quantum computing, quantum algorithms and their applications to computer graphics. ACM SIGGRAPH 2005 Courses. ACM, 2005.

[10] B. L. Douglas and J. B. Wang. A classical approach to the graph isomorphism problem using quantum walks. Journal of Physics A: Mathematical and Theoretical, 41(7), 075303, 2008

[11] D. Emms, R. C. Wilson and E. R. Hancock. Graph matching using the interference of discrete-time quantum walks. Image and Vision Computing, 27(7), 934-949, 2009.

[12] M. Gori M, M. Maggini and L. Sarti. Exact and Approximate Graph Matching Using Random Walks. IEEE Transactions on Pattern Analysis and Machine Intelligence. 27(7): 1100-1111 2005.

[13] D. Emms, R. C. Wilson and E. R. Hancock. Graph matching using the interference of continuous-time quantum walks. Pattern Recognition 42(5): 985-1002 2009

[14] D. Emms, S. Severini, R. C. Wilson, E. R. Hancock. Coined quantum walks lift the cospectrality of graphs and trees. Pattern Recognition 42(9): 1988-2002, 2009

[15] S. Nene, S. Nayar, H. Murase. Columbia Object Image Library. COIL, Technical Report CUCS-006-96. Department of Computer Science, Columbia University, 1996.

[16] C. Harris, M. Stephens, A combined corner and edge detector, in :Pro. of 4th Alvey Vision Conference, Manchester, vol 15, 1988, pp 147-151.