

A NEW ALGORITHM FOR INDUCED SUBGRAPH ISOMORPHISM

Nadia M. G. AL-SAIDI¹, Nuha A. RAJAB², and Hayder N. ABDUL-RAHMAN³

Many algorithms to solve subgraph isomorphism problems have been proposed and proven to be NP-hard, but they did not demonstrate promising results especially on the large and dense graphs. In this paper, a new algorithm for determining an induced subgraph isomorphism between pattern and target graphs is proposed. It is based on decomposition of a graph into components and refinements. The incident matrices are used to help in rearranging the vertices in a descending order and reducing the search efforts. The algorithm is analyzed from complexity point of view to demonstrate its effectiveness after applying it on several types of graphs.

Keywords: graph isomorphism, induced subgraph, incidence matrix, time.

MSC 2000: 05C50, 05C60, 68R10, 05C78.

1. Introduction

The common algorithms for finding subgraph isomorphism are those that are based on backtracking in a search tree, where different algorithms are used to prevent the search tree from large growing. Ullman algorithm [1], is the best known one, and there is also the general graph matching suggested by Cordella et al., [2] is another one. Solving this problem in polynomial time has received a lot of research attention (see, for example, Dessimark et al.[3], Eppstein [4]). The subgraph isomorphism problems that are based on heuristic search techniques are the most interesting techniques in recent researches (see, for example, Akinniyi et al., [5], Cortadella and Valiente [6], Larrosa and Valiente [7]). In practice, the one that considered to be efficient is the algorithm that described by Foggia et al. [8].

This algorithm reduces the computational cost of the matching process by using a set of feasibility rules. VF2 is a procedure introduced by Cordella et al., [9]. It improves Ullmann's refinement by reducing the number of backtracks with the help of a forward checking technique, and thereby reducing the total search

¹ Prof., Branch of Applied Mathematics - Applied Sciences Department - University of Technology -Baghdad-Iraq, e-mail: nadiamg08@gmail.com

² Assoc Prof., Branch of Applied Mathematics - Applied Sciences Department - University of Technology -Baghdad-Iraq, e-mail: nuha.abd10@gmail.com

³ MSC Student, Branch of Applied Mathematics - Applied Sciences Department - University of Technology -Baghdad-Iraq

space by using advanced data structures. Krissinel and Henrick [10], introduces an algorithm to enumerate all possible mappings of subgraph of the two graphs recursively. Finding simple path and cyclic in randomized method is presented by Alon et.al., [11]. Based on building a “plan graph”, a new algorithm for finding a graph-subgraph mapping is described by Betz [12]. A multi vertex matching is introduced also in some papers, (see, for example [13-15]), where a matching is done between a vertex in one graph with a set of vertices in the others. Other general techniques handling graph matching are given also in [16-21]. They proposed a deterministic matching method for verifying both graph and subgraph isomorphism, the search space is explored by means of depth-first search technique, given by Lipest et.al., [22].

During comparison in the networks to report error and noise, numerous error-tolerance mechanisms are introduced to handle the problem of graph and subgraph isomorphism. They are more suitable for real life networks that are essential incorrect and incomplete because they are error-tolerance [23-25]. An emerging approach based on graph database appears to speed up pattern processing. In this approach that adopts filtering and refinement, some subgraph of the target large graph are filtered because they cannot contain pattern graph; Zheng et al., [26] is an example of one of the methods that follows this approach. In addition to the current Section, this paper contains four more Sections that organized as follows: Some notations and definitions are presented in Section 2. Our proposed heuristic search algorithm with a demonstrative example is presented in Section 3. The analysis and discussion for the proposed algorithm are reported in Section 4. Finally, the paper is concluded in Section 5.

2. Notations and Definitions

The common notations and the fundamental definitions used in this work are introduced in this section. For the readers who are interested in more details, we refer them to [27-28].

Definition 2.1 The *neighborhood* (or *open neighborhood*) of a vertex v , denoted by $N(v)$, is the set of the vertices adjacent to v ; $N(v) = \{u \in V(G) \mid (u, v) \in E(G)\}$, and the closed neighborhood is $N[v] = N(v) \cup \{v\}$.

Definition 2.2 The mapping $L: V \rightarrow N$ is called vertex-labeled of a graph G (or simply labeled). The label of a vertex v is given by $L(v)$.

Definition 2.3 A graph $G = (V, E)$ is said to be dense if for every $v \in V$, $\deg(v) > n/2$, where $|V| = n$.

Definition 2.4 If $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$, for any two graphs H and G , then H is said to be a subgraph from G denoted by $H \leq G$.

Definition 2.5 A Graph $H = (V', E')$ is called an induced subgraph of G denoted by $H \trianglelefteq G$ if and only if $E' \subseteq E$ whose endpoints are both in V' .

Definition 2.6 The matrix $Q = (b_{i,j})_{n \times m}$ is called an incidence matrix of the undirected graph G , where n and m are the numbers of vertices and edges respectively, and

$$b_{i,j} = \begin{cases} 1 & \text{if } v_i \text{ is incident on } e_j \\ 0 & \text{otherwise} \end{cases}$$

Definition 2.7 A subset $M_G \subseteq E$ is called a matching of G if no vertex in V is incident on more than one edge in M_G (i.e. no two edges of M_G have a vertex in common). The process of finding vertices and edges correspondence to the graphs $G_P = (V_P, E_P)$, and $G_T = (V_T, E_T)$, that satisfies some constraints is called graph matching such that, it ensures that similar substructures in one graph are mapped to similar substructures in the other.

Definition 2.8 For the graphs $G_P = (V_P, E_P)$ and $G_T = (V_T, E_T)$, there is a subgraph isomorphism from G_P to G_T if there is an injective function $f: V_P \rightarrow V_T$ and a subgraph $S \leq G_T$ such that, f is a graph isomorphism from G_P to S satisfies:

- For all $v \in V_P, f(v) = \dot{v} \in V_T, f^{-1}(\dot{v}) = v$
- For all $e = (v_1, v_2) \in E_P$, there exists a distinct edge $\dot{e} = (f(v_1), f(v_2)) \in E_T$.

Definition 2.9 A subgraph isomorphism from $G_P = (V_P, E_P)$ to $G_T = (V_T, E_T)$, is called an induced subgraph isomorphism if there is an induced subgraph from G_T denoted by G_S , such that $G_P \cong G_S$. In this case, a corresponding bijection between vertices of G_P and G_S is said to be an induced sub-isomorphism between two graphs.

3. The Proposed Algorithm

In this section, a new algorithm is proposed to find an induced subgraph isomorphic between two graphs $G_P = (V_P, E_P)$, which is known as the pattern graph, and $G_T = (V_T, E_T)$, the target graph, such that, G_T is a dense, undirected, and connected graph, with $|V_T| \geq |V_P|$, and G_P is an undirected, connected graph.

The proposed method presents a heuristic algorithm for induced subgraph isomorphism based on decomposing graph into components and refinements to construct identification for induced subgraph isomorphism, such that, each component represents all paths with minimum number of edges between v_i and v_j denoted by $I_{v_i v_j}$.

The structure of the graph in this algorithm is determined depending on the paths and distances between the vertices. The proposed method is applied in four phases algorithm designed as follows:

Phase 1: The labeling of the graph is performed in this part based on the incidence matrix that helps to rearrange the vertices in a descending order according to the degree of the vertices. For a given graph G , the degree of all vertices, $\deg(v_1), \deg(v_2), \dots, \deg(v_n)$, is computed such that, the degree sequence of G is exist, which helps to obtain the incidence matrix in descending order.

Hence, the random search is improved and the searching time is decreased. This can be illustrated in the following steps:

1. Input $|V| = n$, the order of G ; $|E| = m$, the size of G .
2. Input the incident matrix $Q_{n \times m} = [q_{ij}]$.
3. Compute the degree of vertices, $\deg(v_i) = q_{i(m+1)} = \sum_{j=1}^m q_{ij}$, $i = 1, \dots, n$.
4. Rearrange rows of $Q_{n \times m}$ in descending order such that, $\deg(v_{R_1}) \geq \deg(v_{R_2}) \geq \dots \geq \deg(v_{R_n})$.
5. Relabeled the vertices of the given graph, such that, the vertex in the first row is labeled as v_1 and so on.

Phase 2: This part is to determine all the induced subgraphs in the given graphs based on the minimal paths between any two vertices in order to construct the induced matrix $I(u, v)_{n \times n}$, for any two vertices u and v in G , all distances are computed and all paths with minimum length between v_i and v_j are determined and denoted by $I_{v_i v_j}$. This can be illustrated in the following steps:

1. Compute the distance matrix $D_{n \times n} = [\delta_{G_{ij}} = d(v_i, v_j) > 1]$.
2. The induced matrix $I(u, v)_{n \times n}$ for any two vertices u and v is defined by the article $I_{v_i v_j}$ that represents the set of all paths between the vertices v_i and v_j with minimum distance greater than one, such that $I(u, v)_{n \times n} = [d(v_i, v_j)I_{v_i v_j}]$; where,
- $I_{v_i v_j} = \{e_i, I_1 = (v_i, e_{i+1}, \dots, e_j, v_j), I_2 = (v_i, e_k, \dots, e_l, v_j), \dots, I_r = (v_r, e_k, \dots, e_s, v_j)\}$, if $e_i = (v_i, v_j)$, where r is the number of different paths between v_i and v_j .
 $I_{v_i v_j} = \{I_1, I_2, \dots, I_r\}$ otherwise.

Phase 3: This phase is designed to find one or more paths that include all vertices and edges of G that represent its identification. In Phase 1, the vertex v of the largest degree is chosen, and in Phase 2, the set of the minimal paths is found, whereas, in this phase, the set of paths incident on v that contains all vertices and edges in G is determined. If they do not exist, the searching is transformed to the $N(v)$ such that, the chosen vertex u is the vertex of the largest degree. If the chosen paths contains all vertices and edges in G , then the search is stopped; otherwise, continue in this manner until all vertices and edges of a given graph is obtained. Since the given graph is dense. Therefore, the maximum number of search steps equals to the number of the paths incidence on one vertex or at most two vertices. Hence, the searching range is $(n/2, n-1)$. In other words, the identification of the graph is obtained by the following steps.

- a. Search in $I(u, v)_{n \times n}$ about the location (i, j) of the paths that contains a maximum number of vertices and edges to identify the vertex v_i , if $d(v_i, v_j)I_{v_i v_j}$ consist of all the vertices and edges of G , then let $I_G = I_{v_i v_j}$, and go to c.

- b. Search in the incidence matrix $Q_{n \times m}$ about all vertices v_j neighbors to the v_i (i.e. $v_j \in N(v_i)$, $j = 1, 2, \dots, |N(v_i)|$) to rearrange them in descending order according to their degree. Now, check for each v_k , $k = 1, 2, \dots, |N(v_i)|$, if $v_k \in V(I_{v_i v_j})$ then omits this vertex; otherwise find another path $I_{v_i v_k}$ with additional new vertices in the induced matrix to be added with the previous path by union them, such that $I_G = I_{v_i v_j} \cup I_{v_i v_k}$.
- c. Identify the graph $G = (V, E)$ by the chosen I_G .

Phase 4: After the identification of the pattern graph I_{G_P} and target graph I_{G_T} , that performed in phase 3, the induced subgraph isomorphism is accomplished in this final phase by searching the entries of the induced matrix $I_{G_T}(u, v)_{n \times n}$.

Search about all induced subgraphs in G_T , which are isomorphic to G_P , i.e. seeking in $I_{G_T}(u, v)_{n \times n}$, about the identification I_{G_P} . Using the induced matrix in phase 2, to check whether $d_{G_T}(v_i, v_j)I_{G_T v_i v_j}$ is an induced subgraph or not. This can be illustrated in the following steps.

- a. For all $i, j = 1, \dots, n$ do the following;

1. Find the number of distinct vertices, $\alpha_T = \left| \bigcup_{i=1}^{|I_{G_T v_i v_j}|} V(I_{G_T v_i v_j}) \right|$.
2. Find the number of distinct edges $\beta_T = \left| \bigcup_{i=1}^{|I_{G_T v_i v_j}|} E(I_{G_T v_i v_j}) \right|$.
3. The number of paths $\gamma_T = \left| I_{G_T v_i v_j} \right|$, where $I_{G_T v_i v_j}$ is the set of all paths.
4. If $\alpha_T \neq |V(G_P)|$, or $\beta_T \neq |E(G_P)|$, go to b.
5. If $\gamma_T = \gamma_{G_P}$ then $d_{G_T}(v_i, v_j)I_{G_T v_i v_j}$, is an induced subgraph isomorphism.

End for.

- b. Decomposition of $I_{G_T v_i v_j}$,

For all $I_{G_T v_i v_j}$, do the following:

- i. Let $I_{G_T v_i v_j} = \{I_{T_1}, I_{T_2}, \dots, I_{T_{\gamma_T}}\}$.
- ii. For $i, j = 1, 2, \dots, \gamma_T$, $\forall i < j$, define $I_{G_{T_{ij}}}$, such that $I_{G_{T_{ij}}} = \{I_{T_i} \cup I_{T_j}\}$.
If $I_{G_{T_{ij}}} = I_{G_P}$, then $d_{G_T}(v_i, v_j)I_{G_{T_{ij}}}$ is an induced subgraph isomorphism to pattern graph G_P .

End for.

End for.

3.1. Theoretical results

To demonstrate the validity of paths used in search procedure to be minimal paths, and used to represent the induced subgraph that we have been seeking, the following lemma, propositions, and theorem are proven.

Lemma1:

Let $G = (V, E)$ be a simple connected dense graph, and $|V| = n$, then the distance between every non adjacent pair of vertices u and v , is $d(u, v) = 2$ or 3 .

Proof:

Since G is a simple connected dense graph, and $|V| = n$, then $\deg(v) > \frac{n}{2}$, $\forall v \in V$ the distance $d(u, v)$ refer to the length of a minimal path between u and v . If u and v are non-adjacent vertices, and there exist a vertex $w \in N(u) \cap N(v)$, where $N(u)$, and $N(v)$ are the neighborhoods of the vertices u , and v respectively, then it is obvious that $d(u, v) = 2$, because G is a connected graph. If $w \notin \{N(u) \cap N(v)\}$, and $w \in N(u)$, then $\exists x \in V$, such that $x \in \{N(w) \cap N(v)\}$, and $x \notin N(u)$, since $\deg(v) > \frac{n}{2}$, then $d(u, v) = 3$.

If $x \in N(v)$, and $x \notin \{N(u) \cup N(w)\}$, then there exist a vertex y such that $y \in \{N(w) \cap N(x)\}$, that means $d(u, v) = 4$, but this contradiction with $|V| = n$, since $\deg(v) > \frac{n}{2}$, for all $v \in V$, which implies $|V| = u + v + N(u) + N(v) + y = 3 + 2 \frac{n}{2} = n + 3$. Hence, $d(u, v) = 3$. \square

Proposition 1

In a dense, simple, connected graph, every path is an induced subgraph.

Proof:

Since the graph is dense then by Lemma 1, all paths are of length two or three. It is clear that all minimal paths of length two is an induced subgraph, for the paths of length 3, let $p_3 = \{u, e_1, w_1, e_2, w_2, e_2, v\}$, be such a path, neither u and w_2 nor w_1 and v are adjacent, which is a contradiction with length three; then p_3 is an induced subgraph. \square

The following Proposition, determines the lower and upper bounds of the number of paths between two vertices in a dense graph. This will helps to reduce the search space, and to improve the efficiency of the proposed method from the complexity point of view.

Proposition 2

In a dense, simple, connected graph, the number of paths between any two vertices u , and v is greater than $\frac{n}{2}$; and less than $n-1$.

Proof:

To prove the lower bound, for a dense graph, and by using Lemma 1, $\deg(v_i) > \frac{n}{2}$, $i=1,2,\dots,n$, and the length of minimum paths is 2, or 3.

- The first case; when the length of minimum paths is 2, s.t. $\exists w_i \in \{N(u) \cap N(v)\}$. If all paths between u and v is of length two then $|N(u) \cap N(v)| > \frac{n}{2}$, then the number of paths of length 2 is $|N(u) \cap N(v)|$.
- The second case; if the length of the minimum paths is 3, and $\{N(u) \cap N(v)\} = \emptyset$, then there exist two set of vertices $w_i \in N(u)$, and $x_i \in N(v)$, such that $\deg(w_i) = \deg(x_i) > \frac{n}{2}$, also w_i , and x_i are adjacent for all $i=1, 2, \dots, \frac{n}{2}$, then there exist at least $\frac{n}{2}$ different paths of length 3.

To prove the upper bound, when the dense graph is a complete graph, then $\deg(v) = n-1$ for all v , such that, the result is trivial and there exist $n-1$ paths between u and v . \square

Theorem 1

Let $G = (V, E)$ be a simple, connected, dense graph. If the distance between any non-adjacent pair of vertices u and v is a minimum, then the subgraph between u , and v defined on the set of minimal paths of length 2 or 3 is induced.

Proof

Let G be a simple, connected, dense graph, and $P = \{P_1, P_2, \dots, P_t\}$ be a set of paths of length 2 or 3 between any pair of non-adjacent vertices u , and v . Let $S = (V(S), E(S)) \leq G$ be a subgraph contains the set of paths P . To prove that S is induced, the following three cases are proven:

Case (1): If $\bigcap_{i=1}^t V(P_i) = \emptyset$ and $\bigcap_{i=1}^t E(P_i) = \emptyset$ (i.e. all P_i s have distinct vertices and edges), then each path $P_i, i = 1, 2, \dots, t$, is an induced subgraph with the set of vertices $V_i = \{u, v_i, v\}$ because $v_i \notin \bigcap_{i=1}^t V(P_i)$. Also since $\forall e = \{x, y\} \in E(S), \exists x, y \in V(S)$, then the subgraph S on the set vertices $V(P) = \{u, v_1, v_2, \dots, v_t, v\}$ is an induced subgraph.

Case (2): If there exist a vertex v_i such that $v_i \in \{V(P_i) \cap V(P_j)\}$, then there exist an edge $e_i \in \{E(P_i) \cap E(P_j)\}$, such that the subgraph $S = (V(S), E(S))$; where $V(S) = \bigcup_{i=1}^t V(P_i)$; and $E(S) = \bigcup_{i=1}^t E(P_i)$; is an induced subgraph, since all edges incident on $V(S)$ belongs to $E(S)$, for illustration.

Case (3): If the set of vertices are disjoint (i.e. $\bigcap_{i=1}^n V(P_i) = \emptyset$), and there exist an edge $e_i = (x, y)$ such that $x \in V(P_i), y \in V(P_j)$, compute all minimum paths between any pair of the vertices $v, u, x, and y$, where u , and v are the ends of the given path, such that $e = (x, y)$ exist in the induced subgraph that contains these vertices. \square

4. Results and Analysis

In this paper, a new algorithm for finding an induced subgraph isomorphism between two graphs known as pattern and target graphs has been proposed. It is based on searching about all minimal paths between any two vertices to represent

the induced subgraph in one of them that is isomorphic to the other. This is done if the searched path (or paths) between two vertices contains all vertices and edges in their chosen subgraph. Otherwise, additional vertices are added to the path to be self-contained, and in this case, it is considered as a path between more than two vertices. Hence, the resulting induced subgraph is constructed by joining these paths together.

4.1 Comparison results

The algorithm is designated to find the induced subgraph in a target graph isomorphic to the pattern graph. It is divided into four main parts. In the first part, the labeling for both graphs is performed as well as, rearranging the vertices in descending order according to the degree of vertices. This will lead to constricting of a new matrix called induced matrix illustrated in part two, such that, each entry (i,j) represents an induced subgraph of all minimal paths between a pair of vertices (v_i, v_j) . This matrix is used to facilitate the search space about the paths that contains all vertices and edges of the selected subgraph. Therefore, the graphs are identified using these sets of paths. Finally, the last part is designated to find all induced subgraph in a pattern graph that are isomorphic to the target graph by seeking about the identical sets of vertices and edges in both graphs. After applying this algorithm on different cases, some of the concluding remarks are abstracted as follows:

- This algorithm can be used for graph isomorphism and induced subgraph isomorphism.
- Choosing the maximum degree vertex helps to reduce the search space that is better than random search.
- Using this algorithm for finding all induced subgraph is an advantage upon some well-known algorithms designated to find only one graph or subgraph isomorphism.
- Based on minimal path search, it is better and faster than other types of search based on a chosen starting point randomly. This is because the determination of start and end vertices will facilitate the search space and improve the complexity of the search algorithm.
- The density condition on the graph being used for search is another advantage that leads to a wide area of application of the proposed algorithm over some known algorithms suitable to be applied for low connected graphs.
- The worst case arises when the vertices of the given graph is complete, so this case is overtaken in our algorithm.

The proposed algorithm is compared with Ullmann [1] and VF2 [2] algorithms in terms of their properties. This comparison is abstracted in Table 1.

Table 1
Compression between our proposed method and Ullmann and VF2 Methods

	Ullmann	VF2	The proposed method
The implementation	Isomorphism and subgraph isomorphism	Isomorphism and subgraph isomorphism	Isomorphism and induced subgraph isomorphism
Matrix investigation	Adjacency	-----	Distance
Starting point	Random vertex	Determine a partial set of vertices	Vertex of largest degree
Output result	Single graph	Single graph	All induced isomorphic subgraph
Search method	Depth first search	Tree search	All paths
Graph structure	General	General	Dense
Search process	Permute the rows and column	State space representation	Minimal paths between two vertices

5. Conclusions

A new heuristic algorithm for finding all induced subgraph isomorphism between two graphs is proposed. It is based on minimal paths between any pair of vertices in both graphs. The vertices are rearranged in descending order according to the degree of vertices to help in suppressing or preventing the searched path from appearing in other searchers. Some theoretical results are concluded and proved to give some consolidation to the proposed algorithm. A comparison of the properties of the proposed algorithm with the well-known traditional algorithms, Ullmann and VF2 is performed. The possibility of applying the proposed algorithm on a highly connected dense graph helps to expand fields of application that many traditional algorithms failed to cover.

R E F E R E N C E S

- [1]. *J. R. Ullmann*, "An algorithm for subgraph isomorphism," ACM, **vol. 23**, no.1, 1976, pp. 31-42.
- [2]. *L. P. Cordella, P. Foggia, C. Sansone, and M. Vento*, "A (sub) graph isomorphism algorithm for matching large graphs," IEEE transaction on pattern analysis and machine intelligence, **vol. 26**, no.10, 2004, pp. 1367-1372.
- [3]. *A. Dessmark, A. Lingas, and A. Proskurowski*, "Faster algorithms for subgraph isomorphism of k-connected partial k-trees. Algorithmica", **vol. 27**, no.3, 2000, pp.337-347.
- [4]. *D. Eppstein*, "Subgraph isomorphism in planar graphs and related problems", *J. Graph Algorithms & Applications*, **vol. 3**, no. 3, 1999, pp.1-27.
- [5]. *F.A. Akinniyi, A.K.C. Wong, and D.A. Stacey*, "A new algorithm for graph monomorphism based on the projections of the product graph," IEEE, **vol.16**, no.5, 1986, pp.740-751.
- [6]. *J. Cortadella, G. Valiente*, "A relation view of subgraph isomorphism," In proceedings of fifth international seminar on relation methods in computer science, 2000, pp. 45-54.
- [7]. *J. Larrosa, G. Valiente*, "Graph pattern matching using satisfaction," In proceedings of joint Appligraph/Getgrats workshop of graph transformation systems, 2000, pp. 189-196.

- [8]. *L. P. Cordella, P. Foggia, C. Sansone, M. Vento*, “An improved algorithm for matching large graphs”, 3rd IAPR-TC15 Workshop on Graph-based Representations in Pattern Recognition, Cuen, 2001, pp.149-159.
- [9]. *L. P. Cordella, P. Foggia, C. Sansone, and M. Vento*, “Performance evaluation of the vf graph matching algorithm”. In Proceedings of ICIAP 1999, pp. 1999, pp.1172–1177.
- [10]. *E. B. Krissinel and K. Henrick*, “Common subgraph isomorphism detection by backtracking search,” Software Practice and Experience, **vol. 34**, no. 6, 2004, pp.591–607.
- [11]. *N. Alon, R. Yuster, and U. Zwick*. “Color-coding. Electronic Colloquium on Computational Complexity (ECCC),” J.ACM **vol. 42**, no. 4, 1995, pp. 844-856.
- [12]. *G. V. Batz*, “An optimization technique for subgraph matching strategies,” Technical Report 2006-7, Universitat Karlsruhe, Faculty of Informatik, 2006.
- [13]. *R. Ambauen, S. Fischer, and H. Bunke*, “Graph edit distance with node splitting and merging,” IAPR-TC15 WKSP on Graph-based Representation in Pattern Recognition, 2003, pp. 95–106.
- [14]. *M. Boeres, C. Ribeiro, and I. Bloch*, “A randomized heuristic for scene recognition by graph matching,” In Proc. of WEA 2004, 2004, pp. 100–113.
- [15]. *P. Champin and C. Solnon*, “Measuring the similarity of labeled graphs,” Conference on Case-Based Reasoning (ICCBR), 2003, pp. 100–113.
- [16]. *O. Sammoud, C. Solnon, and K. Ghdira*, “An algorithm for the graph matching problem,” In Proceedings of EvoCOP 2005, 2005, pp. 213–223.
- [17]. *S. Zhang, S. Li, and J. Yang. Gaddi*, “Distance index based subgraph matching in biological networks,” In Proceedings of EDBT Conference, 2009.
- [18]. *K. Shuji, O. Yota, S. Toshiki, U. Takeaki*, “Subgraph isomorphism in graph classes,” Discrete Mathematics **vol. 312**, 2012, pp.3164–3173.
- [19]. *W. Tao, D. Guojun, V. De.*, “A polynomial algorithm for submap isomorphism of general maps,” Pattern Recognition Letters, **vol. 32**, 2011, pp.1100–1107.
- [20]. *F. Fedor, L. Daniel, R. Venkatesh, S. Saket, B.V. Raghavendra*, “Faster algorithms for finding and counting subgraphs,” Journal of Computer and System Sciences **vol. 78**, 2012, pp. 698–706.
- [21]. *P. Le Bodica, P. H’erouxb, S. Adamb, Y. Lecourtierb*, “An integer linear program for substitution-tolerant subgraph isomorphism and its use for symbol spotting in technical drawings, or manuscript,” Pattern Recognition, **vol. 45**, no.12, 2012, pp. 4214-4224.
- [22]. *V. Lipets, N. Vanetik, E. Gudes*, “Subsea: An efficient heuristic algorithm for subgraph isomorphism”, Springer, **vol. 19**, 2009, pp. 320-350.
- [23]. *M. S. Amin, A. Bhattacharjee, Jr. R. L. Finley, and H. Jamil*, “A stochastic approach to candidate disease gene subnetwork extraction”, In ACM international symposium on applied computing, (2010a), pp. 1534–1538.
- [24]. *M. S. Amin, A. Bhattacharjee, and H. Jamil*, “A cytoscape based integrative framework for efficient sub-graph isomorphic protein-protein interaction motif lookup”, In ACM international symposium on applied computing, (2010b), pp. 1572–1576.
- [25]. *A. Bhattacharjee, and H. Jamil*, CodeBlast: “A graph matching approach toward computing functional similarity of interacting networks”, Department of Computer Science, Wayne State University, 2011.
- [26]. *W. Zheng, L. Zou, X. Lian, H. Zhang, W. Wang, and D. Zhao*, SQBC: “An efficient subgraph matching method over large, and dense graphs”, Information Sciences **vol. 261**, 2014, pp. 116–131.
- [27]. *J.A. Bondy U.S.R. Murty*, “Graph theory”, Springer, 2008.
- [28]. *R. Diestel*, “Graph theory of Graduate Texts in Mathematics”, Springer-Verlag, Berlin, third edition, 173, 2005.