An Algorithm to Find the Upper Bound of the Distance Between Graphs[†]

Ping Wang and Gerhard W. Dueck
Department of Mathematics and Computing Sciences
St. Francis Xavier University
Antigonish, Nova Scotia, Canada

Abstract

The problem of finding the distance between two graphs is known to be NP-complete. In this paper we describe a heuristic algorithm that uses simulated annealing to find an upper bound for the distance between two graphs. One of the motivations for developing such an algorithm comes from our interest in finding the diameter of families of non-isomorphic extremal graphs. We tested our algorithm on each family of extremal graphs with up to 16 vertices. We show that the exact distance was obtained in all cases.

1. Introduction

The importance of determining whether two graphs are isomorphic is well known. If two graphs are not isomorphic, then it is often useful to know how different they are. Three areas where the concept of maximum common subgraph arises are (1) in the context of algorithmically recognizing the structural features that occur in a chemical reaction [1], (2) determining the maximum commonalties between structures [2], and (3) developing a metric for studying the relationships between molecular structure and the chemical properties [3]. There is an excellent review article on this subject by Chartrand *et al* [4]. We use their definitions as stated below. Two graphs $G_1=(V_1, E_1)$ and $G_2=(V_2, E_2)$ are *isomorphic*, $G_1 \cong G_2$, if and only if there exists a one to one correspondence f: $V_1 \rightarrow V_2$ such that $(u,v) \in E_1$ if and only if $(f(u),f(v)) \in E_2$. A graph G is a *common subgraph* of graphs G_1 and G_2 if and only if there exists H_1 and H_2 such that $H_1 \subseteq G_1$ and $H_2 \subseteq G_2$ and $H_1 \cong G$ and $H_2 \cong G$. A *maximum common subgraph* (MCS) is a common subgraph which contains the maximum number of edges. The distance between two graphs G_1 and G_2, is defined as follows:

$$d(G_1, G_2) = |E_1| + |E_2| - 2|E_{1,2}| + ||V_1| - |V_2||$$

where $|E_{1,2}|$ is the number of edges of a MCS. The *diameter of a family of graphs* is defined as $\text{Diam}(F) = \max\{d(G,H) | G,H \in F\}$. In this paper we only consider the distance between the graphs with same order and size and it follows that $d(G_1, G_2) = 2(|E_1| - |E_{1,2}|)$. Given the mapping $f: V_1 \rightarrow V_2$, we define the distance induced by f as follows:

$$d_f(G_1,G_2) = 2(|E_1| - |E_f|)$$

where $|E_f|$ is the number of edges in a common subgraph of G_1 and G_2.

† Research supported by the Natural Sciences and Engineering Research Council of Canada and by the St. Francis Xavier University Council for Research.

Let $G_1=(V_1, E_1)$, $G_2=(V_2, E_2)$, and k be a positive integer. Consider the decision problem "Do there exist subsets $E_1' \subseteq E_1$ and $E_2' \subseteq E_2$ with $|E_1'| = |E_2'| \geq k$ such that the two subgraphs $G_1'=(V_1, E_1')$ and $G_2'=(V_2, E_2')$ are isomorphic?". This is a NP-complete problem [5]. A linear time algorithm for trees was developed by Edmonds and Matula [5]. In section 2, we describe the simulated annealing technique which we used to develop an algorithm that can find an upper bound on the distance between two graphs.

Another motivation for developing an algorithm to find the distance between two graphs is to find the diameters of families of non-isomorphic extremal graphs without three-cycles and four-cycles. In 1975, P. Erdös [6] mentioned the problem of determining the value of $f(v)$, the maximum number of edges in a graph of order v and girth at least five. Let $F(v)$ be all non-isomorphic extremal graphs of order v and girth at least five. Recently, Garnick *et al.* published two papers on $f(v)$ [7,8]. They employ search techniques, such as hill-climbing and hill-tracking, to find the graphs without 3-cycles and 4-cycles. The lower bounds on $f(v)$ for $v \leq 200$ are listed in [7]. Furthermore, they prove that the lower bounds are the exact values of $f(v)$ for $v \leq 24$. All graphs in $F(v)$ for $v \leq 21$ are enumerated in [8]. Based on their work we try in section 3, to explore more structural properties for each family of graphs by addressing the following two questions: (1) what is the distance between the graphs in $F(v)$, and (2) is there a common subgraph of size $|E|$ - (Diam($F(v)$)/2. In other words, if $d =$ Diam(($F(v)$)/2 is the largest distance between two graphs in $F(v)$, can we find a graph H of size $|E|-d$ such that it is a subgraph of all the graphs in $F(v)$? To investigate these questions, it is essential that we have a computer program that gives us a good upper bound of the distance between two graphs.

2. Simulated Annealing

Simulated annealing is a means of finding good solutions to combinatorial optimization problems [9]. The basic operation in this technique is a *move*. A move is a transition from element of the solution space to another. Each move affects the cost of the current solution. Intuitively, one favours cost decreasing moves, since a solution with minimum, or near-minimum, cost is the objective. However, by allowing only such moves, it is likely that the final solution is a local minimum, rather than the absolute minimum. In order to escape from a local minimum, cost increasing moves must be made.

In simulated annealing, prospective moves are chosen at random. If a move decreases the cost it is accepted. Otherwise, it is accepted with probability $P(\Delta E) = e^{-\Delta E/T}$ where T is the temperature and ΔE is the increase in cost that would result from this prospective move. Initially T is large, and virtually all moves are accepted. Gradually T is decreased, thus decreasing acceptance of cost increasing moves. Eventually the system will reach a state in which very few moves are accepted. In such a state, the system is said to be *frozen*. The sequence of decreasing temperatures is called the annealing schedule. The next temperature is obtained by $T_{n+1} = \alpha T_n$, where a is the cooling rate. Typical values for a are in the range from 0.75 to 0.98.

We developed a computer program that uses simulated annealing to find an upper bound for the distance of two graphs. Our program starts with a random mapping $f{:}V_1 \rightarrow V_2$. A prospective move is found by randomly selecting two vertices, v_i and v_j. A new mapping f' is obtained by interchanging the images of v_i and v_j. The cost of the move is calculated a follows:

$$\text{cost} = d_{f'}(G_1,G_2) - d_f(G_1,G_2).$$

A move with negative cost is always accepted. A move with positive cost is accepted with probability $P(\Delta E) = e^{-\Delta E/T}$. The pseudo code for our implementation is given below. In the inner loop, moves are selected at random. A limited number of moves are accepted at each temperature level. We use $20 * |V|$ as a limit. This means that with larger graphs more moves are accepted. Furthermore, there is a limit for the number of moves attempted at each temperature. For each accepted move we want to attempt no more than 80 moves. Once the maximum number of accepted moves or the maximum number of attempted have been reached the temperature is lowered and a new iteration begins. The process stops if the number of accepted moves has not reached the maximum level (max_moves) for more than a given number of consecutive iterations. That is, we consider the system frozen if less than one in 80 attempted moves is accepted.

```
anneal(G1,G2,best_map)
        temp = initial_temp = 1.0
        cool_rate = 0.95
        map = random mapping
        best_map = map
        max_moves = 20*|V|
        max_attempted_moves = 80*max_moves
        max_frozen = 10
        frozen = 0
        while(frozen ≤ max_frozen)
                moves = attempted_moves = 0
                while((moves ≤ max_moves) and
                        (attempted_moves ≤ max_attempted_moves))
                        increment attempted_moves
                        pick a random move map_ran
                        if the move is accepted
                                map = map_ran
                                increment moves
                                if( d_esp(G1,G2)<d_best_map(G1,G2))
                                        best_map = map
                                end if
                        end if
                end while
                temp = temp * cool_rate
                if(attempted_moves > max_attempted_moves)
                        increment frozen
                else
                        frozen = 0
```

```
                end if
            end while
        end anneal
```

The following variables: initial_temp, cool–rate, max_moves, max_attempted_ moves, and max_frozen could have been set differently. We arrived at the settings as shown in the pseudo code above through experimentation and empirical observation. A change in these variables will affect both the results and the execution time of the program. With the values given above our program found the minimum distance for all graphs with 16 vertices described in the next section The program can find the distance of two graphs with 16 vertices in about 12 seconds of CPU time on a DecStation 5000.

3. Results on distances among the graphs in $F(v)$.

We used the computer program, described in section 2, to find the upper bounds of Diam($F(v)$) for $v \le 21$ (see Table 1). We prove that the upper bounds are indeed the exact value of Diam($F(v)$) for $v \le 16$. We also discuss the problem of finding the largest possible common subgraph in each family of graphs, $F(v)$ for $v \le 21$.

We omit the structure for most graphs in $F(v)$, since they can be found in [8]. In Table 1, the values with a star are the upper bounds of Diam($F(v)$).

$F(4)$ has two members, $K_{1,3}$ and P_3. The greatest common subgraph of $F(4)$ is a path of length two. $F(6)$ has two members, C_5 with a pendant edge, and C_6, and the greatest common subgraph of $F(6)$ is a path of length five.

v	1	2	3	4	5	6	7	8	9	10	11	12	13	14
F(v)	1	1	1	2	1	2	1	1	1	1	3	7	1	4
Diam($F(v)$)	0	0	0	2	0	2	0	0	0	0	2	6	0	6

v	15	16	17	18	19	20	21
F(v)	1	22	14	15	1	1	3
Diam($F(v)$)	0	10	10*	10*	0	0	6*

Table 1. Diameter of Families of Graphs.

Diam($F(11)$)=2.

Clearly, Diam($F(11)$)≥2 since the graphs in this family are non-isomorphic graphs. By using our algorithm, we find the upper bound of the distance between any two graphs among the three graphs is two. Hence, Diam($F(11)$)=2. It is interesting to notice that there is a common subgraph of size 14 among three graphs in $F(11)$. The common subgraph consists of the Petersen graph with the following modifications - an additional vertex is joined to any vertex of the Petersen graph and one edge other than the one we added is removed from the vertex of degree four.

Diam(F(12))=6.

	12a	12b	12c	12d	12e	12f	12g
12a	0	2	2	2	4	4	4
12b		0	2	2	2	2	4
12c			0	2	4	4	4
12d				0	4	4	2
12e					0	4	6
12f						0	4

Table 2. Distance between the graphs in F(12).

With the help of the computer program we obtained Table 2. The largest distance between any two graphs in this family is 6 and it only occurs between 12e and 12g. To show Diam(F(12))=6, we have to prove that 12e and 12g cannot be isomorphic graphs when two edges are removed from each graph. Note that 12g is 3-regular graph. However, there are three vertices of degree four with distance three apart in 12e. Clearly, we have to remove at least three edges from 12e such that the remaining graph has maximum degree three.

Garnick and Nieuwejaar [8] define an equivalence class $q_i(G)$ among the vertices in $v(G)$ such that a vertex u is in $q_i(G)$ if and only if u is in i 5-cycles in G. They claim that in each graph they looked at, if two vertices are in same class then they have same degree. This is an interesting statement because it implies that we can get a finer partition using the number of 5-cycles containing each vertex rather than using vertex degrees. We found a counterexample. In the graph 12a (see Figure 1), both vertices 3 and 4 are in seven 5-cycles but $d(3)$=4 and $d(4)$=3.

Note that distance six between two graphs implies that three edges have to be removed from each graph. There is another interesting fact about this family of graphs. Even though there are only two graphs with distance six apart among the seven graphs of size eighteen in F(12), the family F(12) has no common subgraph graph of size fifteen.

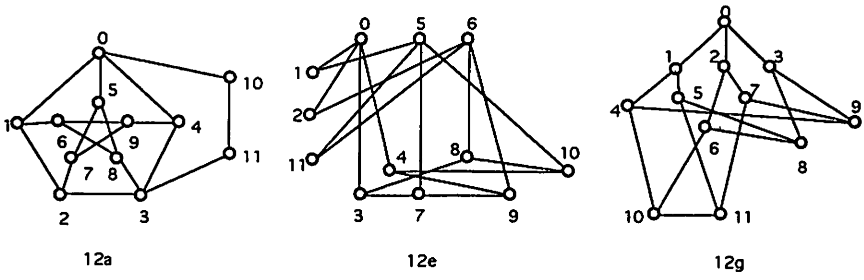


Figure 1. Three graphs from F(12).

Theorem 1. The size of a greatest common subgraph of F(12) is fourteen.

Proof: We will first show that there is no common subgraph of size 15 among 12a, 12e and 12g (See figure 1). Then we find a graph of size fourteen that is a common subgraph of all the graphs in $F(12)$.

Suppose there exists a graph, H, of size 15 such that it is a subgraph of all these three graphs. Let d_i be the number of vertices of degree i for i=0,1,2,3,4. Let $D_2=\{1,2,11\}$, $D_3=\{3,4,7,8,9,10\}$ and $D_4=\{0,5,6\}$ in 12e.

First, we consider the graphs 12e and 12g. The maximum degree of H is three since the maximum degree of 12g is three. This implies that we have to remove one edge from each vertex of D_4 . If two of these three removed edges are joined to the same vertex in D_2, then either 12e has degree sequence $d_0=1$ and $d_1=1$ and this implies we have to remove four edges from 12g, a contradiction, or 12e has degree sequence $d_0=1$, $d_2=3$ and two vertices of degree two at distance two. This contradicts the fact that all three vertices of degree two in 12g must be at least distance three apart. Hence, we have to remove three independent edges from 12e. The three edges cannot be between D_2 and D_4 since it is impossible to have three vertices of degree one by removing three edges from 12g. If two of these three edges are between D_2 and D_4, then we have to remove a path of length three from 12g so there are two vertices of degree one. However, there is a vertex of degree one in D_2 which is distance two from another vertex of D_2 with degree one or two in 12e and there are no such a pair of vertices in 12g, otherwise there is a 3-cycle or 4-cycle (since we remove a path of length three from 12g). Hence, we have to remove either one edge between D_2 and D_4 and two edges between D_3 and D_4, or three edges between D_3 and D_4. In other words, the only possible degree sequences, for the subgraph H are, $d_1=1$, $d_2=4$ and $d_3=7$ or $d_2=6$ and $d_3=6$.

Second, we consider the graphs 12a and 12e where H has a degree sequence $d_1=1$, $d_2=4$ and $d_3=7$. We cannot remove any edge adjacent to vertices 10 and 11 of 12a since this leads to a vertex with degree one adjacent to a vertex with degree two in 12a and it is impossible to have such a pair of vertices in 12e. Note that vertices 10 and 11 are adjacent in 12a, they must correspond to the vertices in D_3 since there is no edge in D_2. Consider the two vertices of degree two in D_3 from which we have removed two edges. They must be at least distance three from D_2 since we remove the only edges between them and D_4. This implies that any vertex with degree one or two in 12a must be at least distance three from vertices 10 and 11. Clearly, we have to remove one edge from 0 and one edge from 3 in 12a since 0 and 3 have degree four. If (0,4) is removed from 12a, then we have to remove (4,3) or else d(4,11)=2. This leads us to remove another edge independent to (0,4) and (4,3) in order to get four vertices with degree two. The two end vertices of the third removed edge in 12a must be at least distance three, since there is no 3-cycle and 4-cycle in 12a and they must correspond with the vertices in D_2 of 12e. This contradicts the fact the distance between any two vertices with degree in D_2 is two. Hence, we have to remove either (0,1) or (0,5) from 12a. Without loss of generality, we may assume that (0,1) is removed. It follows that (1,2) or (1,6) has to be removed in order to have a vertex with degree one in 12a. The removal of (1,2) will result in d(2,11)=2. We have to removed either (3,2) or (3,8) from 3 and in either case there are two vertices with degree one or two corresponding with two vertices in D_2, a contradiction with the fact that D_2 is an independent set.

Finally, we consider the case where d_2=6 and d_3=6. We will use the following two facts about the vertices of degree two in 12e: (a) the six vertices of degree two can be divided into two sets A and B such that A= D_2 contains three independent vertices and B contains those vertices with degree two in D_3; (b): each of the three vertices of set B are at least distance three from A since we have removed the only three edges from B to D_4. Now consider the graph 12a. Since we have to remove one edge from each of 0 and 3, there are two possible ways to remove the third edge: (i) The third edge to be removed is incident with either 0 or 3, say 0. It follows that (0,10) and (3,11) must remain in the graph, and 0, 10 and 11 must be in B since there is no edge among the vertices in A. The vertex of degree two created by the removal of the edge incident with 3 is either distance one from the other two vertices of degree two resulted by the removal of two edges from 0 or distance less than three from 0. This contradicts the facts (a) and (b); (ii) The third edge is independent from the other two edges that have been removed from 0 and 3. Clearly, we cannot remove both (0,4) and (4,3) or else d(4)=1. If one of these two edges, say (0,4), is removed, then 4 is distance 2 from 11 and {4,10,11} must be B. Vertices 1, 5 and 9 are within distance two from {4,10,11} and it follows that we have to remove either (3,8) from 3 and (2,7) as the third edge or (2,3) from 3 and (6,8) as the third edge. However, either d(2,11)=2 or d(8,11)=2 occur in 12a, a contradiction with Fact (b). As a consequence, we have to either remove (0,5) and (2,3) or (0,1) and (3,8) from 12a, since 10 and 11 are in B and A is an independent set. Without loss of generality, we can assume that (0,1) and (3,8) are removed. Since A is a independent set, vertices 10 and 11 must correspond to the vertices in B. Note that vertices 2, 5 and 6 are distance one from 1 or 8. It follows that the edge adjacent to any vertex in {2,5,6} cannot be removed, or else there are two independent edges in the subgraph induced by the vertices of degree two and this implies either there is an edge among the three vertices in 12a are in A or there is a vertex in A adjacent to a vertex in B, a contradiction with facts (a) or (b). Thus, the third edge that has to be removed must be either (7,9) or (9,4). If (9,4) is removed, then {4,10.11} must be set B of 12e and vertex 6 is joined to all three vertices in set A. However, it is easy to see that there is no such vertex in graph 12e. If (7,9) is removed, then vertex 9, with degree two, cannot be in B since it is distance two from vertex 1, a vertex of A, and this contradicts fact (b). Also, it cannot be a vertex in A, or else 6 is the vertex joined to the three vertices in with B, a contradiction.

Through carefully examining all the graphs in this family, we find there is a graph that is the common subgraph of size fourteen in this family of graphs. This subgraph is obtained by removing (0,1), (2,7), (3,4) and (10,11) from 12a. To verify it is indeed a subgraph of all graphs, we run our algorithm on this graph with every graphs in $F(12)$ and the result shows the distance from this graph to every graph in $F(12)$ is eight.

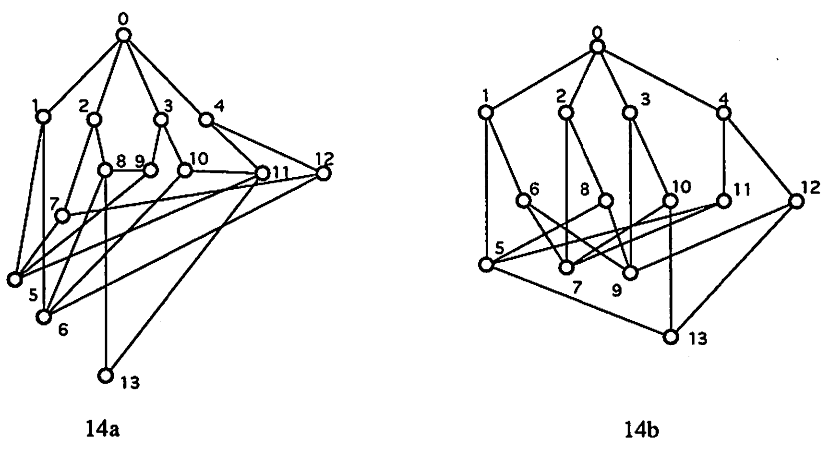
This completes our proof.

Diam(F(14))=6.

	14a	14b	14c	14d
14a	0	4	2	4
14b		0	2	4
14c			0	2

Table 3. Distance between graphs in F(14).

According to our computer program, the upper bound for Diam(F(14)) is four. Therefore, it suffices to show there exist two graphs that are at distance four. We will show that it is not enough to remove one edge from each 14a and 14b (see Figure 2) such that the remaining subgraphs are isomorphic. Note that there are two edges, (5,11) and (6,8), with both end vertices of degree four in 14a and there is no edge among the vertices, {0,5,7,9}, of degree four in 14b. It follows that there must exist an edge between (5,11) and (6,8) in 14a such that there will be no edge among the vertices of degree four after this edge is removed. However, there is no such edge in 14a. Therefore, we have to remove at least two edges from 14a and 14b to obtain isomorphic subgraphs.



14a 14b

Figure 2. Two graphs from F(14).

Again, there is no graph of size twenty-one. Since the proof is similar to the previous case, we omit it. We verified that the graph, H=14a - {(0,3),(2,8),(5,9)}, is a common subgraph of all graphs in F(14) by running our algorithm on this graph with every graphs in F(14).

Diam(F(16))=10

16	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v
a	0	2	4	2	4	10	4	4	2	6	4	4	4	4	4	6	8	6	4	4	6	6
b		0	2	2	2	8	2	6	4	4	2	6	6	6	4	2	4	4	4	6	6	8
c			0	4	2	10	2	4	6	6	4	8	6	6	6	4	8	6	4	8	6	8
d				0	4	10	6	6	6	6	4	4	4	6	4	4	6	4	6	4	4	6
e					0	10	4	4	6	4	2	6	6	6	4	4	6	6	6	6	4	6
f						0	10	8	10	8	8	8	10	8	8	8	4	6	8	8	6	10
g							0	4	6	2	6	6	6	4	6	4	6	4	2	6	4	8
h								0	6	4	4	6	4	4	6	4	4	6	4	6	6	6
i									0	6	2	4	2	4	2	2	6	4	6	2	6	4
j										0	6	4	6	6	6	6	4	4	2	6	6	8
k											0	6	4	6	6	4	4	6	8	4	4	6
l												0	2	2	4	4	4	6	6	2	6	4
m													0	4	2	4	6	4	4	4	6	6
n														0	6	6	4	6	8	4	4	4
o															0	4	6	2	4	6	4	4
p																0	4	2	4	4	4	6
q																	0	2	4	4	2	6
r																		0	2	4	2	6
s																			0	6	4	8
t																				0	4	2
u																					0	4

Table 4. Distance between graphs in F(16).

This is a large family of graphs that contains twenty-two non-isomorphic graphs. As we can see from the above table, the largest distance, ten, between any two graphs only occurs when 16f is one of two graphs. In other words, graph 16f must have some distinct structural feature that the other graphs do not have. We can prove that Diam(F(16))=10 by showing d(16f,16i)=10 [10]. However, the proof is rather long and we omit it here.

In the first examples given in this section, we frequently use the degree sequence to find a common subgraph. It is worth pointing out that the argument on the degree sequence does not work for graphs in general. For example, consider the graphs given in Figure 3. Both graphs, 16f and 16i, have exactly the same degree sequence — eight vertices of degree three and eight vertices of degree four. However, they are nonisomorphic. In fact, five edges have to be removed from each graph before a common subgraph can be found. We cannot follow the trace of degree sequences to find the common subgraph of 16f and 16i. That is, in the process of removing five edges from each graph, the intermediate degree sequences are not necessarily the same.

Before we implemented the algorithm described in this paper, we were not able to find the largest subgraph for 16f and 16i. In fact, this was the primary motivations to develop an algorithm that works quickly for any graph (not just for bipartite graphs or planar graphs) and provides a good estimate for the distance between two graphs. We estimate that an exhaustive search for a common subgraph of 16f and 16i will take about one year of CPU time. Our algorithm finds the distance between 16f and 16i in 16 seconds of CPU time.

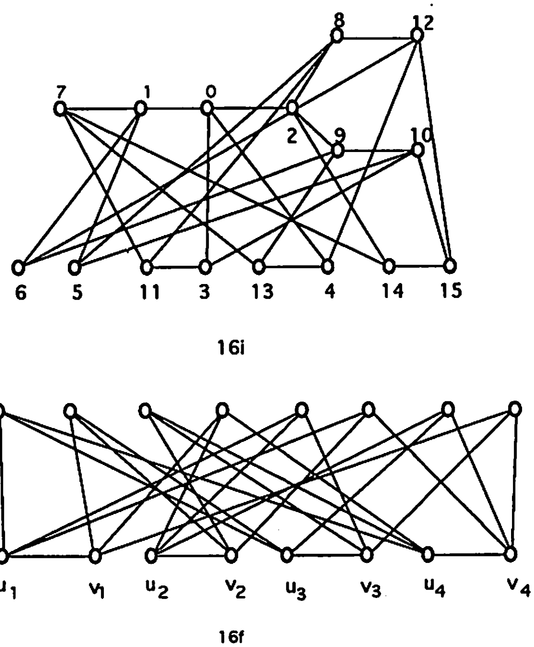


Figure 3. Graphs 16f and 16i from *F(16)*.

4. Concluding Remarks

Garnick *et al.* [7] point out every graph, *G*, without 3-cycles or 4-cycles and with at least five vertices contains $S_{\Delta,\delta-1}$, where Δ (δ) is the maximum (minimum) degree of G. $S_{\Delta,\delta-1}$ is a (Δ,δ)-star, consisting of a vertex called root joined to Δ vertices and each of the Δ neighbors of the root joined to δ-1 additional independent vertices. For example, every graph in F(12) must contain $S_{4,1}$ with eight edges. As shown in Theorem 1, the greatest common subgraph of F(12) has size fourteen. This implies that all graphs in F(12) share some common structure that has more edges than $S_{\Delta,\delta-1}$. This observation is also true for all other graphs we analyzed. In fact, the size of the greatest common subgraph is usually much larger that the size of $S_{\Delta,\delta-1}$. Clearly, two edges can be added to $S_{4,1}$ without creating any 3-cycle or 4-cycle. It will be interesting to find a good estimate of the size of a greatest common subgraph for a family of extremal graphs with girth five.

Another interesting question that arises from investigations is: Under what conditions can a common subgraph for $F(v)$ with size exactly equal to $f(v)$ - (Diam($F(v)$)/2 exist? In other words, is there a graph with the size of a the greatest common subgraph between two graphs that has the largest distance among all pairs of graphs in this family?

Once a large common subgraph in a family has been found, it might be possible to construct more non-isomorphic graphs in $F(v)$ by adding k edges to the known common subgraph, where (Diam($F(v)$)/2 $\leq$ k. For example, the authors in [7] have found two non-isomorphic graphs in $F(26)$. By using our algorithm, we can find a common subgraph (it will not necessary be the greatest common subgraph in $F(26)$) of the only two known members in $F(26)$. Can we add some edges to this common subgraph such that more non-isomorphic graphs of $F(26)$ can be generated?

References

[1] G. E. Vladutz, "Development of a combined WLN/CTR multilevel approach to the logarithmic analysis of chemical reactions in view of their automatic indexing," *British Library Research and Development*, London (1977).

[2] M. M. Cone, R. Verkataraghavan and F. W. McLafferty, "Molecular structure comparison program for the identification of maximal common substructures," J. *Amer. Chem. Soc.* 99, 7668-7671 (1977).

[3] M. A. Johnson, *Relating Metrics, Lines and Variables on Graphs to Problems in Medicinal Chemistry*, John Wiley & Sons Inc., New York, 457-470 (1985).

[4] G. Chartrand, M. A. Johnson, G. Kubicki and O. R. Oellermann, "The theory and applications of greatest common subgraphs," preprint.

[5] M. R. Garey and Johnson, *Computer Intractability: A Guide to the Theory of NP-Completeness*, Freeman, New York (1979).

[6] P. Erdös, "Some recent progress on extremal problems in graph theory," *Congr. Numer.* 14, 3-14 (1975).

[7] D. K. Garnick, Y. H. H. Kwong and F. Lazebnik, "Extremal graphs without three-cycles or four-cycles," *Journal of Graph Theory*, Vol. 17, No. 5, 633-645 (1993).

[8] D. K. Garnick and N. A. Nieuwejaar, "Non-isomorphic extremal graphs without three-cycles or four-cycles," *JCMCC* 12 , 33-56 (1992).

[9] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science*, Vol. 220, pp. 671-680 (1983).

[10] P. Wang and G. W. Dueck, "On the families of extremal graphs without three-cycles and four-cycles," Technical Report TR95-1, St. Francis Xavier University, NS, Canada (1995).