Author: E. Loukakis

Title: A Recursive Algorithm for the Maximum Cardinality Independent set

Creator: HDML

BULLETIN OF THE GREEK MATHEMATICAL SOCIETY
Vol. 21(1980)

A RECURSIVE ALGORITHM FOR THE MAXIMUM CARDINALITY INDEPENDENT SET

By
E. Loukakis

1. Introduction.

A graph $G = (V,E)$ is an ordered pair of a finite non-empty set of $n = |V|$ elements called vertices and of a set $E \subseteq \{(u,v): u,v \in V, u \neq v\}$ of $m = |E|$ unordered vertex pairs called edges. A set $S \subseteq V$ such that $\forall u,v \in S \Rightarrow (u,v) \notin E$ is called **independent** also known as **anti-clique, vertex packing internally stable or simply stable.** Denoting by $\mathcal{I}[G]$ the family of independent sets of a graph $G=(V,E)$, its **independence or stability number** is defined as:

$$\alpha(G) = |S^*| = \max\{|S| : S \in \mathcal{I}[G]\} \qquad \text{(MCIS)}$$

and S^* is a **maximum cardinality independent set.** Let $wgt: V \to R^+ \cup \{0\}$ be a weighting function such that for $u \in V$, we have $wgt(u) \geq 0$ and for $P \subseteq V$, $wgt(P) = \sum_{u \in P} wgt(u)$. Then the graph $G = (V,E)$ is **weighted,** and the **maximum weighted independent set** problem is to find M^* such that

$$wgt(M^*) = \max\{wgt(S): S \in \mathcal{I}[G]\} \qquad \text{(MWIS)}$$

Obviously the stability number is a special case of (MWIS) with $wgt(u) = 1, \forall u \in V$. The (MWIS) can be equivalently formulated as a zero-one integer programming problem of the following form:

$$\text{Max}\{w.x: Qx \leq e_m,\ x \in \{0,1\}^n\}$$

Where $e_m = (1,\ldots,1)$, is an m vector of ones, $w=(wgt(u))$, $u \in V$, is the vector of vertex weights and Q is an mxn edge-vertex incidence matrix of $G = (V,E)$.

Computationally the problems (MCIS) and (MWIS) belong to a class of computationally intractable problems known as NP-Hard, Cook [6], Garey and Johnson [13], Karp [18] and therefore it is most unlikely that a polynomially bounded (Efficient, or Good) algorithm exists or would ever be discovered. There are however, certain classes of graphs for which $\alpha(G)$ is determined by efficent algorithms. Specifically, polynomially bounded algorithms for determining $\alpha(G)$ have been devised by, Burlet and Uhry [4], for parity graphs by Ford and Fulkerson [8] for bipartite graphs, by Hsu et al. [12] for graphs without logn odd cycles, by Fulkerson [9], Garvil [14], [15] for chordal and circular arc graphs and Minty [19] for claw free graphs which include line graphs and comparability graphs. Recently following the publication of the ellipsoid method which is a polynomial algorithm to check the feasibility of a system of linear inequalities by the Russian mathematician L.G. Khachiyah [17], Grötschel, Lovász, and Schrijver [16] exploited these results to obtain among others polynomial algorithms for vertex packing in perfect graphs. Furthermore, the maximum independent set problem can be employed to express a variety of interesting

problems from a number of diverse scientific areas. Some of the applications are referenced in Balas and Samuelson [2], Berge [3] and Houck and Vemuganti [11]. The research appeared so far concerning both the algebraic and algorithmic aspects of maximum independent sets is reviewed in Hansen [10]. To the best of our knowledge none of the algorithms for determining $\alpha(G)$ proposed so far reports conclusive computational experience and the largest graph solved optimally was 100 vertices. Chvâtal [5] and Tarjan and Trojanowski [20] introduced a class of recursive algorithms for determining $\alpha(G)$. Although these algorithms have not been tested computationally, their worst case computational bounds are far better than total enumeration and can therefore considered to be the best among those proposed so far. In this paper we present a recursive algorithm for the maximum cardinality independent set and which we call MCIS . Extensive computational experience on more than 1800 randomly generated graphs ranging from 100 to 250 vertices and from 20% to 90% densities has shown that MCIS is very effective. The remaining parts of the paper are organized as follows. In section 2 we derive theoretical results. These results are exploited in section 3 for the development of MCIS. Section 4 deals with the implementation of the algorithm and presents the computational experience. In appendix the workings of MCIS are illustrated with a numerical example solved in detail. Finally for those researchers who are interested in comparing the performance of their algorithms with that of MCIS, listing of the computer programs of MCIS coded in FORTRAN V can be obtained from the author on request.

2. Dominance and Decompositions.

The following graph and set theoretic concepts and notation will be employed. Let $G = (V,E)$ be a graph. Then for $u \in V$ and $S \subseteq V$, $A(u) = \{v \in V : (u,v) \in E\}$, and $A(S) = \bigcup_{u \in S} A(u)$ will be the adjacency vertex sets of u and S respectively. The subgraph of $G = (V,E)$ induced by the set $K \subseteq V$ is defined as:

$$G(K) = (K, E(K)) = \{(u,v) \in E : u,v \in K\}$$

For any $Q \in \mathcal{J}[G]$, $F(Q)$ is defined as:

$$F(Q) = V \setminus (Q \cup A(Q)) \qquad (1)$$

Where, for any vertex set X and Y, $X-Y \equiv X \setminus Y \equiv X \cap \bar{Y}$ and $\bar{X}$ complements X.

As a direct consequence of (1) we have $A(F(Q)) \cap Q = \emptyset$, $A(Q) \cap F(Q) = \emptyset$ and therefore $Q \cup \{k\} \in \mathcal{J}[G]$ for $k \in F(Q)$. A clique of a graph $G = (V,E)$ is a set $K \subseteq V$ such that $(u,v) \in E$ $\forall\, u,v \in K$ or equivalently $A(u) \cap K = K \setminus \{u\}$, $\forall\, u \in K$. A vertex sequence $P = \{u_1, u_2, \ldots, u_r\}$ of a graph $G = (V,E)$ such that $(u_j, u_{j+1}) \in E$ for $j = 1,2,\ldots,r-1$, is called a path. A vertex set $K \subseteq V$ is connected if $\forall\, u,v \in K$ there exists a path P connecting them and $P \cap K = P$. A connected vertex set S such that R is disconnected $\forall\, R \supset S$ is called a connected component. Thus a connected component is a maximal subgraph.

A maximum cardinality independent set of the graph $G = (V,E)$ and of its subgraph $G(S)$ will be denoted by B^* and $Bst(G(S))$ respectively. $Bst(G(S))$, $S \subseteq V$ is monotonic in the following sence. For any S, $R \subseteq V$ we have.

$$|Bst(G(R))| \leq |Bst(G(S))| \quad \forall\, R \subseteq S$$

Dominance relations.

Dominance relations and the associated tests provide

sufficient conditions that a specific subgraph cannot con - tain a maximum independent set. The value of diminance re - lations rests on the fact that the degree of convergence to the optimum of an algorithm depents directly on the bounding strength and easiness of implementation of the dominance tests employed.

Lemma 1. *Let $G = (V,E)$ be a graph and $u \in V$. Then an independent set which does not contain $\{u\} \cup A(u)$ cannot be maximum i.e.* $|B^*| > |Bst(G(V \setminus (\{u\} \cup A(u))))|$.

The proof of this lemma employs (a) and (b) below which follow directly from the monotonicity property of $Bst(\cdot)$

$$|Bst(G(V \setminus \{u\}))| > |Bst(G(V \setminus (\{u\} \cup A(u))))| \qquad (a)$$

$$|Bst(G(V \setminus A(u)))| > |Bst(G(V \setminus (\{u\} \cup A(u))))| \qquad (b)$$

Combining lemma 1 with the fact that $S \subseteq V$ is independent if $S \cap A(S) = \emptyset$, i.e. S cannot contain both, vertex u and vertex set $A(u)$, we have the following:

Lemma 2. *Let $G = (V,E)$ be a graph and $u \in V$. Then,*

$$a(G) = \max\{|\{u\} \cup Bst(G(V \setminus (\{u\} \cup A(U))))|,\ |Bst(G(V \setminus \{u\}))|\}.$$

Theorem 1. *Let $G = (V,E)$ be a graph, and $u \in V$. Then $A(u)$ is a clique implies $|B^*| \geq |Bst(G(V \cup \{u\}))|$. In other words. If $A(u)$ is a clique then there exists a maximum independent set which contains u.*

The proof of this theorem involves the sequential combined application of lemma 2 and the monotonicity of $Bst(\cdot)$.

Specifically let $A(u) = \{u_1, u_2, \ldots, u_k\}$. Then by lemma 2 we have,

$$a(G) = \max\{|M_0|, |M_1|\}$$

where $M_0 = \{u\} \cup Bst(G(V \setminus (\{u\} \cup A(u))))$ *and*

$M_1 = Bst(G(V \setminus \{u\}))$

Furthermore, for $u_1 \in A(u)$ *we have* $|M_1| = \max\{|M_2|, |M_3|\}$ *and where* $M_2 = \{u\} \cup Bst(G(V \setminus (\{u_1\} \cup A(u_1))))$ *and* $M_3 = Bst(G(V \setminus \{u, u_1\}))$. *However,* $A(u)$ *is a clique implies* $A(u) \subseteq A(u_j)$, $\forall u_j \in A(u)$. *Combining the last with the monotonicity of* $Bst(\cdot)$ *we have* $|M_0| \geq |M_1|$ *and therefore* $\alpha(G) = \max\{|M_0|, |M_2|\}$.

Following the same line of reasoning and for $j = 3, \ldots, k-1$, *letting* $M_j = \{u_j\} \cup Bst(G(V \setminus (\{u_j\} \cup A(u_j))))$ *and* $M_{j+1} = Bst(G(V \setminus \{u, u_1, \ldots, u_j\}))$ *we have*

$$|M_0| \gtrless |M_r| \quad \text{for } r \text{ odd} \qquad (*)$$

and finally $\alpha(G) = \max\{|M_0|, |M_k|\}$ *where* $M_k = Bst(G(V \setminus \{u, u_1, u_2, \ldots, u_k\})) = Bst(G(V \setminus (\{u\} \cup A(u))))$. *Thus,* $|M_0| = |M_k| + 1$ *and the theorem is proved.*

Obviously the inequality (*) *holds as equation for all* u_j *such that* $A(u) = A(u_j)$. *Thus, we obtain the following result:*

Corollary 1. *Let* $G = (V, E)$ *be a graph,* $u \in V$, $A(u)$ *is a clique and* $Q = \{v : A(u) = A(v)\}$. *Then there are* $|Q|$ *alternative maximum independent sets.*

As another immediate consequence of theorem 1 we have:

Corollary 2. *Let* $G = (V, E)$ *be a graph,* $S \in \mathcal{J}[G]$ *and* $u \in S$. *If* $F(s) \cap A(u)$ *is a clique then any independent set which contains* S *and does not contain* u *cannot be maximum.*

An independent set S *of a graph* $G = (V, E)$ *is called* <u>*maximal*</u> *if* $\forall T \supset S$, $T \notin \mathcal{J}[G]$. *From this definition it follows that an independent set which is not maximal cannot be maximum. In* [19] *the following maximality condition has been proved.*

Lemma 3. *Let* $G = (V, E)$ *be a graph,* $S \in \mathcal{J}[G]$ *and* $u \in F(S)$. *If* $A(u) \subseteq A(S)$ *then none of the independent sets* $S \cup \{v\}$,

$v \in F(S) \setminus \{u\}$ can be maximal.

Decompositions.

Let $G = (V,E)$ be a graph. Then, by decomposition we define the family of sets $\{V_1, V_2, \ldots, V_r\}$ such that $V_k \subseteq V$, $k = 1,\ldots,r$, $V_i \cap V_j = \emptyset, \forall\, i \neq j, i,j = 1,\ldots,r$, and $\alpha(G) = |Bst(G(V_1))| + |Bst(G(V_2))| + \ldots + |Bst(G(V_r))|$ and $B^* = Bst(G(V_1)) \cup Bst(G(V_2)) \cup \ldots\ldots \cup Bst(G(V_r))$.

It must be noted that the condiition $V = V_1 \cup \ldots \cup V_r$ is not necessary. These forms of graph decomposition are important for at least the following two reasons:

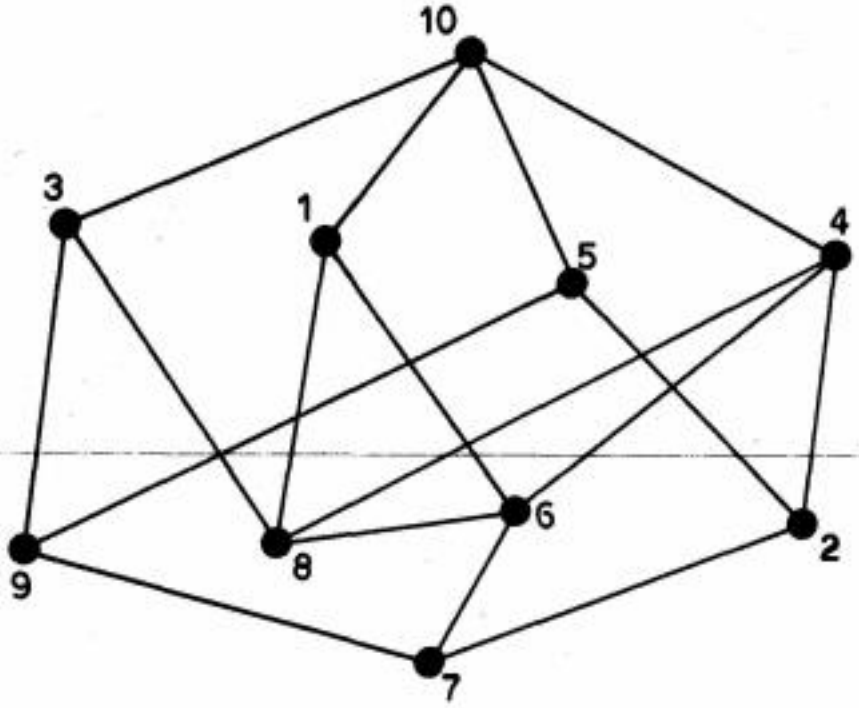
(i) They result in substantial computational savings. Specifically let n be the number of vertices of the graph $G = (V,E)$ which is decomposed into the subgraphs $G(V_1)$, $G(V_2),\ldots,G(V_d)$. Suppose that we apply an algorithm on $G = (V,E)$ to solve the associated (MCIS) problem in time $b.2^n$. Then by applying the same algorithm to the subgraphs $G(V_j)$, $j = 1,\ldots,d$ and then composing the solutions obtained the time required is reduced to $d.b.2^{n/d}$.

(ii) On page 1 we stated the classes of graphs for which polynomially bounded algorithms have been devised. Thus, a graph which does not belong to any of these clas - ses may be solved by a polynomial algorithm in decomposed form provided the decomposing subgraphs belong to these clas- ses. As an illustration consider the graph of figure 1 which does not belong to any of the classes of graphs for which good algorithms exist. Applying lemma 2 its stabili- ty number is computed as:

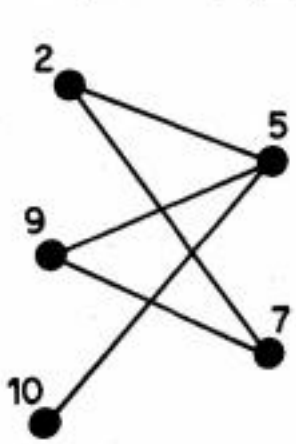
$$\alpha(G) = \max\{|\{8\} \cup Bst(G(V_1))|,\ |Bst(G(V_2))|\}.$$

where $V_1 = \{2,5,7,9,10\}$ and $V_2 = \{1,2,3,4,5,6,7,9,10\}$. However, both subgraphs $G(V_1)$ and $G(V_2)$ are bipartite and can therefore be solved by a polynomial algorithm. Specifical-

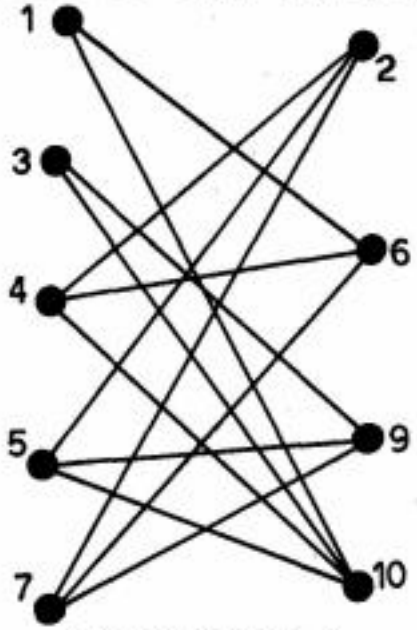
ly the vertex set V_1 can be decomposed into the disjoint independent sets $Q_1=\{2,9,10\}$ and $Q_2=\{5,7\}$ and V_2 into $P_1=\{1,3,4,5,7\}$, and $P_2=\{2,6,9,10\}$. The subgraphs $G(V_1)$ and $G(V_2)$ are illustrated in figure 1.



Graph G = (V,E) decomposed into $G(V_1)$ and $G(V_2)$



Subgraph $G(V_1)$



Subgraph $G(V_2)$

Figure 1.

In the sequel we present decompositions I and II. The study of further forms of decompositions combined with dominance relations as well as their algorithmic consequences will be the subject of a following up paper.

Decomposition I.

A vertex u of the graph G = (V,E) is called a cutvertex or articulation point if the subgraph $G(V \setminus \{u\})$ has more connected components than G = (V,E). Let the cutvertex u of G = (V,E), $V_1, V_2, \ldots, V_k \subset V \setminus \{u\}$, $A(u) \cap V_j \neq \emptyset$, for $j = 1,2,\ldots,k$, and.

$$V_1 \cup V_2 \cup \ldots \cup V_k = V \setminus \{u\},\ V_i \cap V_j \neq \emptyset,$$
$$\forall\ i \neq j,\ i,j = 1,2,\ldots,k \qquad (P)$$

Then:

$$\alpha(G) = \max\{|\{u\} \cup Bst(G(V_1 \setminus (A(u) \cap V_1))) \cup Bst(G(V_2 \cap A(u)))$$
$$\cup \ldots \cup Bst(G(V_k \setminus (A(u) \cap V_k)))|,\ |Bst(G(V_1)) \cup$$
$$\cup Bst(G(V_2)) \cup \ldots \cup Bst(G(V_k))|\} \qquad (D1)$$

The validity of (D1) follows directly from lemma 1 and partition (P). This decomposition is illustrated in figure 2 for the vertex sets $V_1, V_2, V_3 \subset V \setminus \{u\}$. Specifically.

$$\alpha(G) = \max\{|\{u\} \cup Bst(G(Q_1)) \cup Bst(G(Q_2)) \cup Bst(G(Q_3))|,$$
$$|Bst(G(V_1)) \cup \ldots \cup Bst(G(V_2)) \cup Bst(G(V_3))|\} \text{ where}$$
$$Q_1 = V_1 \setminus (V_1 \cap A(u)),\ Q_2 = V_2 \setminus (V_2 \cap (A(u)),$$
$$Q_3 = V_3 \setminus (V_3 \cap A(u)) \text{ and } Q_1 \cap Q_2 \cap Q_3 = \emptyset.$$

To see the huge computational savings resulting from decompositions one may notice that finding the independence number of the graph in figure 2 requires a worst -case time bound of $(p.2^{25})$ where p is a polynomial. However by

plying decomposition I the worst-case time bound is reduced to $O(p2^s)$.

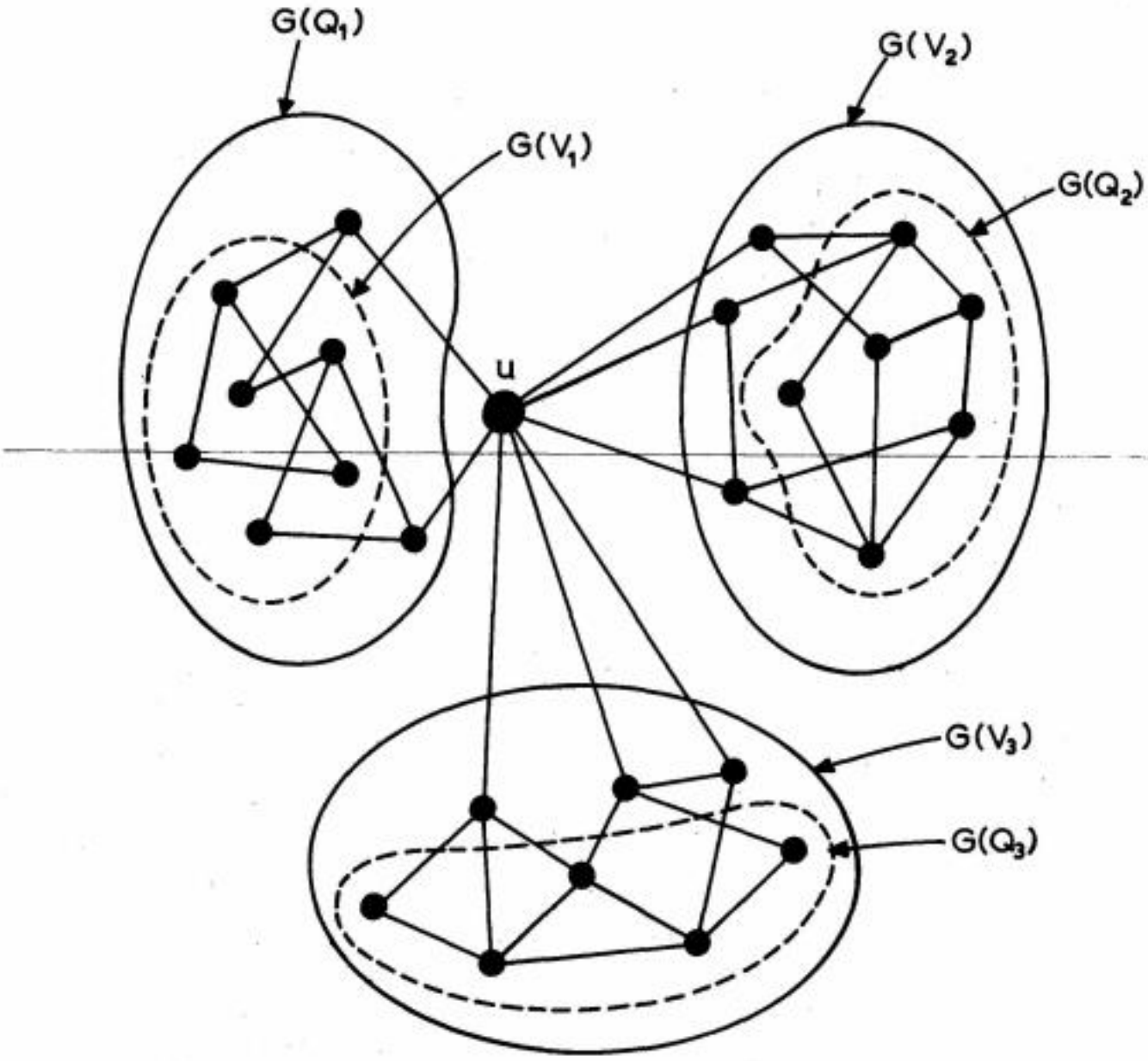


Figure 2

Decomposition II. Consider the graph $G=(V,E)$ with $V_1, V_2 \subset V$, and let, $K = \{u_1, u_2, \ldots, u_r\} \subseteq V_1$ be a clique, $K \subseteq A(v)$ for $v \in V_1$, $A(v) \cap V_2 \neq \emptyset$, ,$A(u) \cap V_2 \neq \emptyset$, $\forall\, u \in K$, and $V_2 \cap (A(V_1) \setminus K) \neq \emptyset$.

Then $\alpha(G)$ is computed as:

$$\alpha(G) = \max\{|Bst(G(P))|, |Bst(G(Q))|\}$$

where $P = V \setminus (\{v\} \cup A(v))$ and $Q = V \setminus \{v\}$.

$$Bst(G(P)) = Bst(G(R)) \cup Bst(G(S)) \qquad (D2)$$

where $R = V_1 \setminus (\{v\} \cup (A(v) \cap V_1))$ and $S = V_2 \setminus (A(v) \cap V_2)$

The validity of (D2) is directly derivable from the following implication

$$V_2 \cap (A(V_1) \setminus K) = \emptyset \wedge K \subseteq A(v) \Rightarrow R \cap S = \emptyset.$$

$$Bst(G(Q)) = \max\{|\{u_1\} \cup Bst(G(Q_1)) \cup Bst(G(S_1))|, |\{u_2\} \cup Bst(G(Q_2)) \cup Bst(G(S_2))|, \ldots, |\{u_r\} \cup Bst(G(Q_r)) \cup Bst(G(S_r))|, |Bst(G(W)) \cup Bst(G(V_2))|\} \qquad (D3)$$

where $Q_k = V_1 \setminus (\{u_k\} \cup (A(u_k) \cap V_1))$ and $S_k = V \setminus (A(u) \cap V)$ for $k = 1,2,\ldots r$, and $W = V_1 \setminus (\{v\} \cup K)$.

One may notice that if $A(v) = K$ the part $Bst(G(W)) \cup Bst(G(V_2))$ should be dropped by the dominance relation of Lemma 1. To prove the validity of (D3) we combine the assumptions of K being a clique and $V_2 \cap (A(V_1) \setminus K) = \emptyset$ which imply $W \cap V_2 = \emptyset$ and $Q_j \cap S_j = \emptyset$, for $j = 1,\ldots,r$.

This form of decomposition is illustrated in figure 3 for the vertex sets $V_1, V_2 \subset V$ and $K = \{u_1, u_2\}$. Specifically.

$$\alpha(G) = \max\{|\{v\} \cup Bst(G(P))|, |Bst(G(Q))|\}$$

where $P = V_1 \setminus \{v, u_1, u_2, u_3, u_4\}$ and $Q = V_2 \setminus \{v_1, v_2\}$.

Further $|Bst(G(Q))| = \max\{|\{u_1\} \cup Bst(G(Q_1)) \cup Bst(G(S_1))|, |\{u_2\} \cup Bst(F(Q_2)) \cup Bst(G(S_2))|, |Bst(G(W)) \cup Bst(G(V_2))|\}$.

where $Q_1 = V_1 \setminus \{v, u_1, u_2, u_3, u_5\}$, $S_1 = V_2 \setminus \{v_4\}$.

$Q_2 = V_1 \setminus \{v, u_1, u_2, u_3, u_6\}$, $S_2 = V \setminus \{v_3, v_5\}$,

$W = V_1 \setminus \{v, u_1, u_2\}$ and V_2. The subgraphs $G(V_1)$, $G(V_2)$, $G(P)$ and $G(Q)$ are illustrated only in figure 3. One may no-

tice again that the worst-case time bounds for the graph of figure 3 is $(p.2^{10})$ and $(p.2^{20})$ with and without decompositions respectively. However, we believe that computational savings resulting from decompositions their greatest value lies in the fact that the decomposing subgraphs may belong to the classes for which polynomially bounded algorithms exist. Specifically if we devise a decomposing technique by which any graph can be decomposed into polynomially many subgraphs which belong to the classes for which polynomially bounded algorithms exist, the NP-comleteness problem has been solved.

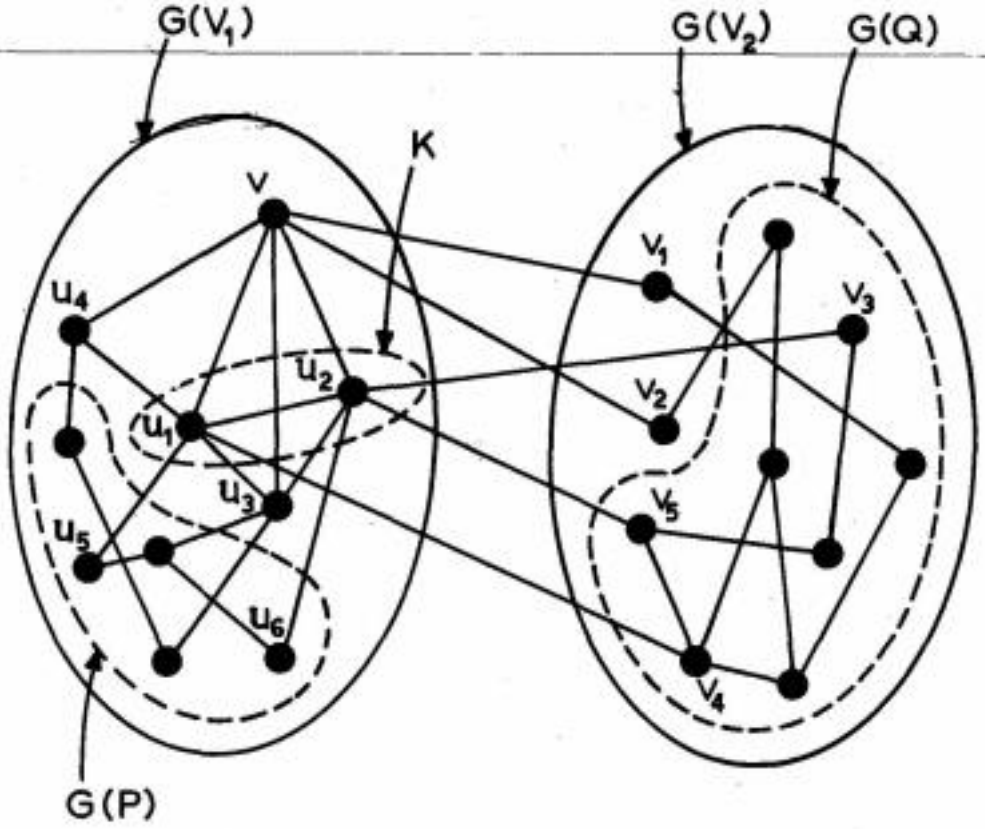


Figure 3

Conclusively graph decompositions reduce the computation time drastically and at least enlagre the class of NP-

hard (complete) problems for which polynomial algorithms are applicable.

3. Development of the algorithm

The algorithm MCIS employs a recursive backtracking scheme by which the family $\mathcal{J}[G]$ is sequentially partitioned according to lemma 2 and searched. Each node of the solutions tree generated is completely specified by the triple $Q = (Q^+, Q^-, F(Q))$ where $Q^+ \in \mathcal{J}[G]$, $F(Q) = V \setminus (Q^- \cup Q^+ \cup A(Q^+))$, $Q^- \cap Q^+ = \emptyset$. In words Q^+ is an independent set of branched vertices, Q^- is the set of backtracked vertices and $F(Q)$ is the set of free vertices such that $Q^+ \cup \{j\} \in \mathcal{J}[G]$, $\forall j \in F(Q)$. When we branch we augment Q^+ with the smallest vertex of $F(Q)$, thus, aiming to find an independent set of greater cardinality than the one already known and which contains every element of Q^+ and none of Q^-. Branching is followed by the process of bounding. This involves the application of various tests which provide sufficient conditions that the augmentation of an independent set by elements of a vertex subset cannot produce a maximum independent set. MCIS employs the following bounding tests which are based on the results of the previous section.

B1 Let $Q = (Q^+, Q^-, F(Q))$ and $q \in F(Q)$. If $A(g) \cap F(Q)$ is a clicque then according to corrolary 1 the nodes of the solution tree which contain Q^+ and donnot contain q cannot produce an independent set of greater cardinality than those which contain q. In this case the node of the solution tree specified by $Q = (Q^+, Q^- \cup \{q\}, F(Q) \setminus \{q\})$ and all its descendants are excluded from further consideration.

B2 Let $Q = (Q^+, Q^-, F(Q))$ and $u \in F(Q)$. If $A(u) \cap F(Q) = \emptyset$, then according to lemma 3 the nodes of the solution tree specified by the triple Q which contain Q^+ and donnot con-

tain u cannot lead to a maximum independent set and are therefore dropped from further consideration.

B3 Let $Q = (Q^+, Q^-, F(Q))$ and $\tilde{Q}$ an independent set known at a specific stage of computations. The it is obvious that the inequality $|Q^+| + |F(Q)| \leq \tilde{Q}$ provides sufficient conditions that any augmentation of Q^+ by subsets of $F(Q)$ can only produce independent sets of smaller cardinality than that of $\tilde{Q}$. In this case the node of the solution tree specified by Q is dropped from further consideration.

When a node of the solutions tree is bounded the process backtracks by removing from Q^+ the vertex which caused the bounding (This is the most recently introduced element in Q^+) and placing it in Q^-. The above analysis is systematized in the steps of algorithm MCIS.

Algorithm MCIS

Step 1. (Initialize) Set $Q^+ = \emptyset$, $Q^- = \emptyset$, $F(Q) = V$, $\tilde{Q}$ = any known independent set and proceed to 2.

Step 2. (Branch, Revise the best known independent set). If $F(Q) = \emptyset$ set $\tilde{Q} \leftarrow Q^+$, and go to 6. Otherwise set $m \leftarrow \min\{j : j \in F(Q)\}$, and proceed to 3.

Step 3. (Apply B1, B2). If $A(m) \cap F(Q)$ is a clique or $A(m) \cap F(Q) = \emptyset$, label m (set $m \leftarrow m^*$) and proceed to 4. Otherwise, proceed to 4.

Step 4. (Update). Set $Q^+ \leftarrow Q^+ \cup \{m\}$, $F(Q) \leftarrow F(Q) \setminus ((\{m\} \cup A(m)) \cap F(Q))$ and proceed to 5.

Step 5. (Apply B3) If $|Q^+| + |F(Q)| \leq |\tilde{Q}|$ proceed to 6 otherwise go to 2.

Step 6. (Backtrack, Terminate). If either $Q^+ = \emptyset$ and $|F(Q)| \leq |\tilde{Q}|$ or every vertex of Q^+ is labelled terminate. Otherwise let u be the right most unlabelled vertex of Q^+ and $R(u) = \{j \in Q^+ : j > u\}$.

Set $Q^+ \leftarrow Q^+ \setminus (R(u) \cup \{u\})$, $F(Q) \leftarrow F(Q) \cup (A(R(u)) \setminus A(Q^+))$, $Q^- \leftarrow Q^- \cup \{u\}$, and go to 5.

The detailed application of MCIS is illustrated with a graph of 10 vertices in appendix 1. To find an initial independent set Q we devised a once through greedy algorithm which we call GREMAX and which is comprised of the following steps.

Step 1. Set $\tilde{S} = V$, $\tilde{Q} = \emptyset$ and proceed to 2.

Step 2. Find j^* such that $n(j^*) = \min\{n(j)=|A(j)\cap\tilde{S}|: j\in\tilde{S}\}$ and set $\tilde{Q} \leftarrow \tilde{Q} \cup \{j^*\}$, $\tilde{S} \leftarrow \tilde{S} \setminus (\{j^*\} \cup A(j^*))$.

If $\tilde{S} = \emptyset$ terminate. Otherwise repeat 2.

GREMAX has been tested on 100 randomly generated graphs and was found to perform extremely well. Specifically the solution return by GREMAX was between 92% to 100% of the optimum return by MCIS.

4. Implementation and computational experience.

In writting this section we tried to follow as closely as possible the guidlines suggested by Crowder et al. [7]. To store the graph and implement MCIS we employed various forms of linked lists data structure Aho, et al. [1]. The compu - ter memory requirements were $2|E|+6|V|+1$ computer words. The performance of MCIS was tested on more than 1300 randomly generated graphs ranging from 100-250 nodes and from 20% to 90% densities. The results produced are presented in ta - bles I and II below and correspond to sets of 10 gnaphs each. To generate the graphs randomly we devised a subroutine called RANGRU and the random numbers generated were drawn from a uniform distribution by calling the subroutine RANDU available in UNIVAC. Before MCIS was executed the nodes of the graphs were ordered according to the relation $i < j \Leftrightarrow |A(i)| \leq |A(j)|$. To relabel the nodes of the graph we devised a subroutine called GISOM which constructs a graph

TABLE I

Number of vertices	Density 20%			Density 30%			Density 40%			Density 50%		
	Low	Avg.	High	Low	Avg.	High	Low	Avg.	High	Low	Avg.	High
100	8.35	12.45	16.42	6.31	10.28	14.17	5.62	7.39	11.62	4.15	6.25	10.81
110	13.42	19.33	22.50	10.22	14.32	20.29	9.23	12.43	14.25	6.32	10.47	13.29
120	16.35	26.81	34.21	14.35	20.51	28.08	12.46	15.60	20.81	8.92	12.13	17.89
130	21.17	32.26	45.30	20.13	27.12	35.44	16.25	21.87	29.42	11.21	16.35	22.68
140	39.10	50.45	62.84	31.25	38.45	50.12	22.50	30.45	40.24	17.48	22.84	31.45
150	44.83	58.61	80.26	40.92	46.32	61.38	33.68	41.38	52.73	24.62	31.28	40.93
160	61.07	76.44	91.42	75.44	80.91	90.42	57.72	59.10	73.87	40.24	46.52	54.27
170	84.42	98.35	118.35	81.38	98.42	110.25	69.83	67.40	94.09	51.53	59.87	72.61
180	118.19	131.18	152.10	105.42	120.85	131.48	80.49	94.23	112.68	62.86	80.66	94.82
190	167.35	180.22	211.22	140.26	162.50	185.74	112.62	120.47	146.83	83.54	105.24	120.43
200	200.68	218.62	246.27	180.68	210.87	222.18	140.27	165.78	190.30	110.20	140.17	158.06
210	204.25	240.38	269.12	192.45	225.38	239.62	171.08	193.26	217.20	131.49	162.34	174.25
220	231.17	260.72	294.25	216.20	242.65	263.24	194.20	212.69	232.51	160.50	180.40	203.72
230	255.10	278.59	313.84	230.08	266.30	294.40	210.75	240.06	260.44	183.57	210.87	224.80
240	289.26	310.48	351.25	250.32	298.43	325.20	231.23	261.70	295.68	210.25	235.18	251.23
250	312.41	335.22	374.39	285.44	310.25	348.17	245.86	280.29	310.57	222.84	260.29	284.30

TABLE II

Number of vertices	Density 60%			Density 70%			Density 80%			Density 90%		
	Low	Avg.	High	Low	Avg.	High	Low	Avg.	High	Low	Avg.	High
100	3.90	5.13	8.36	3.40	4.28	6.84	3.15	3.90	6.04	3.10	3.64	5.88
110	5.22	7.84	10.29	4.13	6.36	8.17	3.88	5.17	7.66	3.45	4.27	6.93
120	7.46	9.36	14.86	6.27	7.20	11.25	4.92	6.84	9.18	4.00	5.88	8.14
130	8.94	14.78	19.43	7.81	10.87	15.40	6.45	9.32	14.25	6.12	8.76	11.38
140	14.27	19.06	28.17	10.39	14.22	22.37	9.16	12.48	20.36	7.85	10.24	18.42
150	20.82	28.32	34.56	14.55	21.73	29.43	12.81	17.25	24.48	10.34	16.32	23.58
160	31.63	40.93	48.81	23.48	34.62	40.15	20.48	29.58	36.83	16.83	27.81	34.27
170	43.25	54.12	62.08	32.59	44.81	53.27	30.62	38.72	49.54	28.52	36.45	42.81
180	50.87	71.88	81.37	44.68	59.68	72.74	41.37	49.77	60.20	36.45	45.38	54.26
190	70.04	89.46	98.74	61.73	70.50	81.33	56.28	66.85	73.89	51.88	60.65	69.93
200	91.89	110.82	132.81	78.45	92.46	112.45	69.84	84.70	100.60	62.36	72.74	86.14
210	112.44	144.57	145.60	94.27	114.28	131.23	88.07	102.16	120.44	81.38	93.25	110.48
220	136.38	168.23	183.25	123.58	150.72	164.39	114.38	135.68	153.25	100.52	119.83	142.69
230	170.50	189.12	206.50	145.37	174.25	185.46	131.28	152.60	169.32	123.67	140.37	155.70
240	195.43	206.84	228.45	168.72	183.37	208.32	152.87	171.87	190.70	139.25	156.85	184.25
250	214.39	231.50	253.18	193.16	212.56	225.67	180.76	193.45	205.40	170.87	188.40	190.76

$\tilde{G} = (\tilde{V},\tilde{E})$ isomorphic to $G = (V,E)$. Incorporating the graph isomorphism subroutine GISOM in MCIS has resulted in improved CPU times in the area of 30%.

References

1 A.V. Aho, J.E. Hopcroft and J.D. Ullman, "The Design and Analysis of Computer Algorithms" (Addison-Wesley, Reading MA., 1974).

2 E. Balas and H. Samuelson "Finding a Minimum Node Cover in an Arbitary Graph" MSRR No 235 Carnegie Mellon University Pittsburg (1973).

3 C. Berge "Graphs and Hypergraphes"(North Holland,1973).

4 M. Burlet and J. P. Uhry. "Parity Graphs". Mathematiques Appliques et Informatique. Rapport De Recherche No 232 1981 Universite Scientifique et Medicale et Institut National Polytechnique de Grenoble.

5 V. Chvătal. "Determining the Stability Number of a Graph". SIAM on Computing 6 (1977), 643-662.

6 S. Cook. "The Complexity of Theorem-Proving Procedures", Proceedings of the Third ACM Symposium on Theory of Computing, Association for Computing Machinery New York 1970), 151-158.

7 H. P. Crowder, R.S. Dembo, and J.M. Mulvey. "Reporting Computational Experiments in Mathematical Programming" Mathematical Programming 15 (1978), 316-329.

8 G.W. Ford and D.R. Fulkerson "Flows in Networks"(Princeton University Press, Princeton 1962).

9 D.R. Fulkerson. "Anti-Blocking Polyhedra" Journal of Combinatorial Theory 12 (1972), 50-71.

10 P. Hansen. "Bornes et Algorithmes pour les Stables d' une Graphe". in Regards sur la Theorie des Graphes eds P. Hansen and D. de Werra (Presses Polytech - niques Romandes 1980).

11 D.J. Houck and R.R. Vemuganti. "An Algorithm for the Vertex Packing Problem" Operations Research 25 (1977) 773-787.

12 W. Hsu, Y. Ikura and G.L. Nemhauser. "A Polynomial Algorithm for Maximum Weighted Vertex Packings on Graphs without Long Odd Cycles". Mathematical Programming 20 (1981), 225-232.

13 M.R. Garey and D.S. Johnson. "Computers and Intractability: A Guide to the Theory of NP-Completeness. (Freeman 1979).

14 F. Garvil. "Algorithms for Minimum Coloring, Maximum Clique, Minimum Covering by Cliques, and Maximum Independent Set of a Chordal Graph" SIAM on Computing 1 (1972), 180-187.

15 F. Garvil. "Algorithms for Circular-Arc Graphs" Net - works 4 (1975), 357-369.

16 M. Grötschel, L. Lovâsz and A. Schrijver "The Ellip - soid Method and its Consequences in Combinatorial Optimization" WP 80151-OR. Institut Für Okonome - trie und Operations Research Universität Bonn(1980)

17 L.G. Khachiyan (Hacijan) "A polynomial Algorithm in Linear Programmin", Doklady 244 (No 5, Feb. 1979) 1093-1096. Translated as Soviet Mathematics Dokla- dy 20, 191-194.

[18] R. Karp "Reducibility Among Combinatorial Problems" in Complexity of Computer Computations. R.E.Miller and J.W. Thatcher eds. (Plenum Press. New York 1972), 85-104.

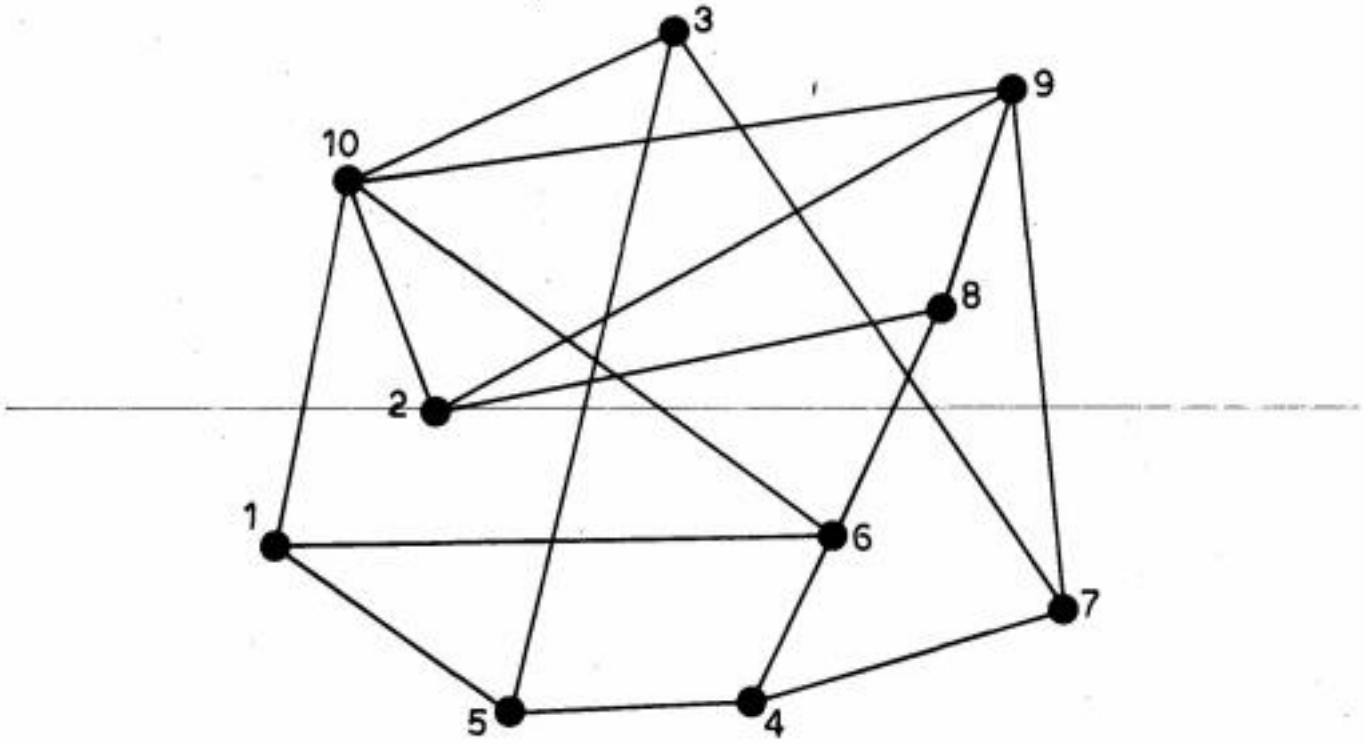
[19] E. Loukakis and C. Tsouros. "A Depth Frist Search Algorithm to Generate the Family of Maximal Independent Sets Lexicographically" to appear in Computing.

[20] G.J. Minty. "On Maximal Independent Sets of Verti - ces in Claw-free Graphs". Journal of Combinatorial Theory (B) 28 (1980), 284-304

[21] R.E. Tarjan and A.E. Trojanowski "Finding a Maximum Independent Set, SIAM on Computing 6 (1977), 537-546.

Appendix

The algorithms MCIS and GREMAX are employed to find a maxium independent set of the following graph.



Vertex j)A(j)

1) {5,6,10} 2) {8,9,10} 3) {5,7,10} 4) {5,6,7}
5) {1,3,4} 6) {1,4,8,10} 7) {3,4,9} 8) {2,6,9}
9) {2,7,8,10} 10) {1,2,3,6,9}

First the GREMAX to obtain a good starting independent set $\tilde{Q}$.

Steps

1) Set $\tilde{S}$ = {1,2,3,4,5,6,7,8,9,10}, $\tilde{Q}$ = ∅

2) $j^* = 1$. Thus set $\tilde{Q} \leftarrow \{1\}$, and $\tilde{S} \leftarrow \{2,3,4,7,8,9\}$
2) $j^* = 3$. Thus set $\tilde{Q} \leftarrow \{1,3\}$, and $\tilde{S} \leftarrow \{2,4,8,9\}$
2) $j^* = 4$. Thus set $\tilde{Q} \leftarrow \{1,3,4\}$, and $\tilde{S} \leftarrow \{2,8,9\}$
2) $j^* = 2$. Thus set $\tilde{Q} \leftarrow \{1,3,4,8\}$, and $\tilde{S} \leftarrow \emptyset$. Terminate.

MCIS.

1) Set $Q^+ \leftarrow \emptyset$, $F(Q) \leftarrow \{1,2,3,4,5,6,7,8,9,10\}$, and proceed to 2.
2) Since $F(Q) \neq \emptyset$, set $m \leftarrow 1$ and proceed to 3.
3) Since $A(1) \cap F(Q) = \{5,6,10\}$ does not form a clique and $A(1) \cap F(Q) \neq \emptyset$ go to 4.
4) Set $Q^+ \leftarrow \{1\}$, $F(Q) \leftarrow \{2,3,4,7,8,9\}$ and go to 5.
5) Since $|Q^+| + |F(Q)| = 7 > |\tilde{Q}| = 4$ go to 2.

2) Since $F(Q) \neq \emptyset$, set $m \leftarrow 2$ and go to 3.,
3) Since $F(Q) \cap A(2) = \{8,9\}$ is a clique set $2 \leftarrow 2^*$ and go to 4.
4) Set $Q^+ \leftarrow \{1,2^*\}$, $F(Q) \leftarrow \{3,4,7\}$ and go to 5.
5) $|Q^+| + |F(Q)| > |\tilde{Q}|$. Thus, go to 2.
2) $F(Q) \neq \emptyset$. Thus set $m \leftarrow 3$ and go to 3.
3) Since $F(Q) \cap A(3) = \{7\}$ is a clique set $3 \leftarrow 3^*$ and proceed to 4.
4) Set $Q^+ \leftarrow \{1,2^*,3^*\}$, $F(Q) \leftarrow \{4\}$ and go to 5.
5) Since $|Q^+| + |F(Q)| = |\tilde{Q}| = 4$ go to 6.
6) $Q^+ \neq \emptyset$ and not every vertex of Q^+ is labelled. Thus, set $u \leftarrow 1$, $R(1) \leftarrow \{2,3\}$, $Q^+ \leftarrow \emptyset$, $Q^- \leftarrow \{1\}$, $F(Q) \leftarrow \{2,3,4,5,6,7,8,9,10\}$ and go to 5.
5) Since $|F(Q)| + |Q^+| > |\tilde{Q}|$ go to 2.
2) Since $F(Q) \neq \emptyset$, set $m \leftarrow 2$ and go to 3.
3) Since $A(2) \cap F(Q) = \{8,9,10\}$ is not a clique and $A(2) \cap F(Q) \neq \emptyset$ go to 4.
4) Set $Q^+ \leftarrow \{2\}$, $F(Q) \leftarrow \{3,4,5,6,7\}$ and go to 5.
5) Since $|Q^+| + |F(Q)| > |\tilde{Q}|$ go to 2.

2) $F(Q) \neq \emptyset$. Thus set $m \leftarrow 3$ and go to 3.

3) Since $F(Q) \cap A(3) \neq \emptyset$ and is not a clique go to 4.

4) Set $Q^+ \leftarrow \{2,3\}$, $F(Q) \leftarrow \{4,6\}$ and go to 5.

5) Since $|F(Q)| + |Q^+| = |\tilde{Q}| = 4$ go to 6.

6) $Q^+ \neq \emptyset$ and not every vertex of Q^+ is labelled. Thus, set $u \leftarrow 3$, $R(3) \leftarrow \emptyset$, $Q^+ \leftarrow \{2\}$, $F(Q) \leftarrow \{4,5,6,7\}$, $Q^- \leftarrow \{1,3\}$ and go to 5.

5) Since $|F(Q)| + |Q^+| > |\tilde{Q}|$ go to 2.

2) $F(Q) \neq \emptyset$. Thus, set $m \leftarrow 4$ and go to 3.

3) $F(Q) \cap A(4) = \{5,6,7\}$ is not a clique. Thus go to 4.

4) Set $Q^+ \leftarrow \{2,4\}$, $F(Q) = \emptyset$ and go to 5.

5) Since $|Q^+| + |F(Q)| < |\tilde{Q}|$ go to 6.

6) $Q^+ \neq \emptyset$ and not every vertex of Q^+ is labelled. Thus, set $u \leftarrow 4$, $R(4) \leftarrow \emptyset$, $Q^+ \leftarrow \{2\}$, $F(Q) \leftarrow \{5,6,7\}$, $Q^- \leftarrow \{1,3,4\}$ and go 5.

5) Since $|Q^+| + |F(Q)| = |\tilde{Q}| = 4$ go to 6.

6) $Q^+ \neq \emptyset$ and not every vertex of Q^+ is labelled. Thus set $u \leftarrow 2$, $Q^+ \leftarrow \emptyset$, $F(Q) \leftarrow \{3,4,5,6,7,8,9,10\}$, $Q^- \leftarrow \{1,2,3\}$, and go to 5.

5) Since $|Q^+| + |F(Q)| > |\tilde{Q}|$ go to 2.

2) Since $F(Q) \neq \emptyset$, set $m \leftarrow 3$ and go to 3.

3) $F(Q) \cap A(3) = \{5,7,10\}$ is not a clique. Thus go to 4.

4) Set $Q^+ \leftarrow \{3\}$, $F(Q) \leftarrow \{4,6,8,9\}$ and go to 5.

5) Since $|Q^+| + |F(Q)| > |\tilde{Q}|$ go to 2.

2) Since $F(Q) \neq \emptyset$, set $m \leftarrow 4$, and go to 3.

3) $F(Q) \cap A(4) \neq \{6\}$. Thus, set $4 \leftarrow 4^*$ and go to 4.

4) Set $Q^+ \leftarrow \{3,4^*\}$, $F(Q) \leftarrow \{8,9\}$ and go to 5.

5) $|F(Q)| + |Q^+| = |\tilde{Q}| = 4$. Thus, go to 6.

6) Since $Q^+ \neq \emptyset$ and not every vertex of Q^+ is labelled set $u \leftarrow 3$, $R(3) \leftarrow \{4\}$, $Q \leftarrow \emptyset$, $F(Q) \leftarrow \{4,5,6,7,8,9,10\}$, Q^- $\{1,2,3\}$, and go 5.

5) Since $|F(Q)| + |Q^+| > |\tilde{Q}|$ go to 2.

2) $F(Q) \neq \emptyset$. Thus set $m \leftarrow 4$ and go to 3.

3) $F(Q) \cap (4) = \{5,6,7\}$ and is not a clique. Thus, go to 4.

4) Set $Q^+ \leftarrow \{4\}$, $F(Q) \leftarrow \{8,9,10\}$ and go to 5.

5) Since $|F(Q)| + |Q^+| = |\tilde{Q}|$ go to 6.

6) $Q^+ \neq \emptyset$ and not every vertex of Q^+ is labelled.Thus set
$u \leftarrow 4$, $R(4) \leftarrow \emptyset$, $F(Q) \leftarrow \{5,6,7,8,9,10\}$ Q^+ $\emptyset$,
$Q^+ \leftarrow \{1,2,3,4\}$ and go to 5.

5) Since $|F(Q)| + |Q^+| > |\tilde{Q}|$ go to 2.

2) $F(Q) \neq \emptyset$. Thus, set $m \leftarrow 5$, and go to 3.

3) Since $A(5) \cap F(Q) = \emptyset$, set $5 \leftarrow 5^*$ and go to 4.

4) Set $Q^+ \leftarrow \{5^*\}$, $F(Q) \leftarrow \{6,7,8,9,10\}$, and go to 5.

5) Since $|F(Q)| + |Q^+| > |\tilde{Q}|$ go to 2.

2) Since $F(Q) \neq \emptyset$, set $m \leftarrow 6$ and go to 3.

3) $A(6) \cap F(Q) = \{8,10\}$, is not a clique. Thus, go to 4.

4) Set $Q^+ \leftarrow \{5^*,6\}$, $F(Q) \leftarrow \{7,9\}$ and go to 5.

5) $|Q^+| + |F(Q)| = |\tilde{Q}|$. Thus, proceed to 6.

6) $Q^+ \neq \emptyset$ and not every vertex of Q^+ is labelled.Thus, set
$u \leftarrow 6$, $R(6) \leftarrow \emptyset$, $Q^+ \leftarrow \{5\}$, $F(Q) \leftarrow \{7,8,9,10\}$,
Q^- $\{1,2,3,4,6\}$ and go to 5.

5) Since $|Q^+| + |F(Q)| > |\tilde{Q}|$ go to 2.

2) $F(Q) \neq \emptyset$. Thus set $m \leftarrow 7$. and go to 3.

3) $F(Q) \leftarrow A(7) = \{9\}$. Thus set $7 \leftarrow 7^*$ and go to 4.

4) Set $Q^+ \leftarrow \{5^*,7^*\}$, $F(Q) \leftarrow \{8,10\}$ and go to 5.

5) Since $|Q^+| + |F(Q)| = |\tilde{Q}|$ go to 6.

6) Since every element of Q^+ in labelled terminate. A maximum independent set is $Q = \{1,3,4,8\}$.

Section of Operations Research
Polytechnii school
University of Thessaloniki
Greece