ON THE MAXIMUM-WEIGHT CLIQUE PROBLEM

by

Egon Balas
Carnegie-Mellon University, Pittsburgh

Vašek Chvátal
McGill University, Montreal

Jaroslav Nešetřil
Charles University, Prague

Carnegie-Mellon University

PITTSBURGH, PENNSYLVANIA 15213

GRADUATE SCHOOL OF INDUSTRIAL ADMINISTRATION

WILLIAM LARIMER MELLON, FOUNDER

This document has been approved for public release and sale; its distribution is unlimited.
ON THE MAXIMUM-WEIGHT CLIQUE PROBLEM

by

Egon Balas
Carnegie-Mellon University, Pittsburgh

Vašek Chvátal
McGill University, Montreal

Jaroslav Nešetřil
Charles University, Prague

June 1985

This document has been approved for public release and sale; its distribution is unlimited.

The research of the first author was supported by Grant ECS-8205425 of the National Science Foundation and Contract N00014-82-K-0329 NR047-607 with the U.S. Office of Naval Research. Reproduction in whole or in part is permitted for any purpose of the U.S. Government.
ABSTRACT

We introduce several new classes of graphs on which the maximum-weight clique problem is solvable in polynomial time. Their common feature, and the central idea of our algorithms, is that every clique of any of our graphs is contained in some member of a polynomial-sized collection of induced subgraphs that are complements of bipartite graphs. Our approach is quite general, and might conceivably yield many other classes of graphs along with corresponding polynomial time algorithms.

Additional keywords: Triangle-free graphs, triangle-free graphs; theorems.
0. INTRODUCTION

The maximum-weight clique problem, or MWCP for short, goes as follows: given a graph whose vertices carry numerical weights, find a clique (that is, a set of pairwise adjacent vertices) whose total weight is as large as possible. This problem is notoriously hard, even when all the weights are equal; the problem of deciding whether a prescribed graph contains a clique of a prescribed size is NP-complete; in fact, this problem was one of the five prototypes of NP-complete problems presented by Cook (1971) in the classical paper that laid the foundations of NP-completeness theory.

Nevertheless, there are several known polynomial-time algorithms each of which solves the MWCP on all graphs coming from some restricted class. In this paper, we present new algorithms of this kind. The method used to design these algorithms is quite general, and might conceivably yield many other algorithms.

1. THE RESULTS

We reserve the letter \( n \) for the number of vertices of a graph \( G \). If \( C \) is a class of graphs and \( t \) is a number, we say that \( C \) is \( t \)-bounded if every \( G \) in \( C \) has \( O(n^t) \) maximal cliques. (Here, as usual, "maximal" is meant with respect to set inclusion, not size.) We say that \( C \) is tame if no \( G \) in \( C \) contains an induced subgraph that is an odd antihole (defined as the complement of an odd hole, i.e. of a chordless cycle whose length is odd and at least five). We say that a green/red coloring of the edges of a graph is \( C \)-formative if the green graph belongs to \( C \) and the red graph contains no triangle. Our key result goes as follows.
THEOREM 1. For every t-bounded and tame class C there is an algorithm that, given any graph G along with a C-formative coloring of its edges, solves any MWCP on G in $O(n^{2t+3})$ steps.||

Our prime example of a t-bounded and tame class of graphs is the class TR of triangulated graphs, defined as graphs in which every cycle of length at least four has a chord. These graphs were introduced by Hajnal and Surányi (1958) and studied further by Berge (1960), Dirac (1961), and many others. In particular, Dirac proved that every triangulated graph has a simplicial vertex, defined as a vertex whose neighbours are all adjacent to each other; an instant corollary of this theorem states that every triangulated graph has at most n maximal cliques. Hence TR is 1-bounded; since every odd antihole contains a chordless cycle of length four or five, TR is also tame.

Now let $C_1$ stand for the class of all graphs that admit C-formative edge colorings. Theorem 1 provides a polynomial-time algorithm for solving any MWCP on any graph in some subclass $C^*$ of $C_1$ if and only if a polynomial-time algorithm to find a C-formative edge coloring of every graph in $C^*$ is available. In particular, if such an algorithm is available for every graph in $C_1$, that algorithm can be used to test membership in $C_1$. Our next result shows that the existence of such an algorithm is unlikely when $C = TR$.

THEOREM 2. Testing membership in $TR_1$ is an NP-complete problem.||

Thus we are led to look for some proper subclass $C^*$ of $C_1$ with a polynomial-time algorithm to construct a C-formative edge coloring of every graph in $C^*$. One way of doing this is to impose additional constraints on the C-formative edge colorings. For instance, rather than requiring that the red graph contain no triangle, we might insist that it be bipartite. This idea leads to the following notion: we say that a partition of the vertex set of G into disjoint parts $V_1$ and $V_2$ is C-formative if each of the two subgraphs of G
induced by $V_1$ and $V_2$ belongs to $C$. Clearly, as long as $C$ is closed under disjoint unions, each $C$-formative vertex partition of $G$ yields a $C$-formative edge coloring of $G$. In this case, Theorem 1 has an instant collary with "coloring of its edges" replaced by "partition of its vertices". In fact, the assumption that $C$ is tame can be dropped, and the resulting statement holds true even if $C$ is not closed under disjoint unions.

**THEOREM 3.** For every $t$-bounded class $C$ there is an algorithm that, given any graph $G$ along with a $C$-formative partition of its vertices solves any MWCP on $G$ in $O(n^{2t+3})$ steps.\|

Let $C^2$ stand for the class of all graphs that admit $C$-formative vertex partitions. We have observed that $C^2 \subseteq C^1$ whenever $C$ is closed under disjoint unions; in particular, $TR^2 \subseteq TR^1$. Examples of graphs in $TR^2$-$TR$ are all holes and antiholes. For holes this is obvious. For an antihole, let $V_1$ and $V_2$ be the odd-numbered and even-numbered vertices, respectively, of the cycle whose complement is the given antihole. Then $V_1$ and $V_2$ is a $TR$-formative vertex partition of the antihole. Examples of graphs in $TR^1$-$TR^2$ are all graphs that contain no triangles and whose chromatic number exceeds four. (Triangle-free graphs of an arbitrarily high chromatic number were constructed first by Tutte, writing under the name of Blanche Descartes (1954); another family of such graphs was constructed independently by Mycielski (1955). For a strong result on such graphs, see Erdős (1959) and Lovász (1968).) Trivially, every such graph $G$ belongs to $TR^1$ (color all edges red). To see that $G \notin TR^2$, observe that every triangulated induced subgraph of $G$ is a forest; since the chromatic number of $G$ exceeds four, two such subgraphs cannot cover all the vertices of $G$. 
Again, Theorem 3 provides a polynomial-time algorithm for solving any \( MWCP \) on any graph in some subclass \( C^* \) of \( C^2 \) if any only if a polynomial-time algorithm to find a \( C \)-formative vertex partition of every graph in \( C^* \) is available. Again, if such an algorithm is available for every graph in \( C^2 \), then the algorithm can be used to test membership in \( C^2 \). Again, we have a result showing that the existence of such an algorithm is unlikely when \( C = TR \).

**THEOREM 4.** Testing membership in \( TR^2 \) is an NP-complete problem.||

To obtain a subclass \( C^* \) of \( C^2 \) such that a \( C \)-formative vertex partition of every graph in \( C^* \) can be found in polynomial time, we may begin with any polynomial-time heuristic that attempts to construct the partition, and then simply define \( C^* \) as the class of those graphs on which the heuristic succeeds. (The same approach can of course be used to obtain a subclass of \( C^1 \) such that a \( C \)-formative edge coloring of every graph in the subclass can be found in polynomial time.) We use this approach with heuristics based on a certain subroutine that we call GREEDY. The input of GREEDY is any graph \( G \) whose vertices have been labeled as \( v_1, v_2, \ldots, v_n \); its output is either a \( C \)-formative vertex partition of \( G \) or a failure message. In the description of GREEDY and later on, we let \( G(S) \) denote the subgraph of \( G \) induced by \( S \).

**GREEDY:**

\[
\begin{align*}
V_1 & \leftarrow \emptyset, \quad V_2 \leftarrow \emptyset \\
\text{for } i = 1, 2, \ldots, n \text{ do} & \\
\text{if } G(V_1 \cup \{v_i\}) \in C \text{ then } & V_1 \leftarrow V_1 \cup \{v_i\} \\
& \text{else } V_2 \leftarrow V_2 \cup \{v_i\} \\
\text{endif} & \\
\text{endfor} & \\
\text{if } G(V_2) \in C \text{ then return } V_1 \text{ and } V_2 & \\
& \text{else return a failure message endif}
\end{align*}
\]
Note that GREEDY runs in time $O(n^{s+1})$ whenever membership in $C$ can be tested in $O(n^s)$ steps for some constant $s$.

Since GREEDY works with labeled graphs, it gives rise to a variety of algorithms working with unlabeled graphs: each of these algorithms first labels the vertices and then applies GREEDY. In the remainder of this section, we shall discuss three special cases in detail.

First, the simplest way to construct a labeling of the vertices of $G$ is to take an arbitrary labeling; we shall let $C^3$ denote the class of graphs on which the resulting algorithm always delivers a $C$-formative vertex partition. To put it differently, $G$ belongs to $C^3$ if and only if GREEDY succeeds on $G$ for each of its $n!$ labelings.

**THEOREM 5.** Let $C$ be any $t$-bounded class of graphs such that members of $C$ can be recognized in $O(n^{2t+2})$ steps. Then any MWCP on any graph in $C^3$ can be solved in $O(n^{2t+3})$ steps.\[\]

Note that the hypothesis of Theorem 5 is satisfied when $C$=TR and $t=1$: Dirac's theorem implies at once that triangulated graphs can be recognized in $O(n^4)$ steps. (Actually, the running time of an algorithm designed by Rose, Tarjan, and Lueker (1976) to recognize triangulated graphs is only $O(n^2)$, but $O(n^4)$ is good enough for our purpose.)

**COROLLARY 5A.** Any MWCP on any graph in $TR^3$ can be solved in $O(n^5)$ steps.\[\]

Our second algorithm attempts to construct a labeling whose properties guarantee the success of GREEDY. If such a labeling is found then GREEDY is applied; else a failure message is returned. To explain the details, we need two more definitions. First, a vertex $v$ in a graph $F$ will be called **C-acceptable** if $F-v$ contains no disjoint sets $S_1, S_2$ of vertices such that

$$F(S_1) \in C, F(S_2) \in C, F(S_1 \cup \{v\}) \notin C, F(S_2 \cup \{v\}) \notin C.$$
Second, a labeling $v_1, v_2, ..., v_n$ of the vertices of $G$ will be called
C-formative if each $v_k$ is C-acceptable in $G((v_1, v_2, ..., v_k))$. Clearly, if
GREEDY is given a graph $G$ with a C-formative labeling, then it finds a
C-formative vertex partition of $G$. (Here, we are tacitly assuming that graphs
with no vertices at all belong to $C$.) The following algorithm, given any
graph $G$, will either find a C-formative labeling of $G$ or establish that no
such labeling exists.

LABEL:

$F \leftarrow G$, $k \leftarrow n$, failure $\leftarrow$ false

While $k > 0$ and failure $= false$ do

if $F$ has a C-acceptable vertex $v$

then $v_k \leftarrow v$, $F \leftarrow F - v$, $k \leftarrow k - 1$

else failure $\leftarrow$ true

endif

endwhile

if $k = 0$ then return $v_1, v_2, ..., v_n$

else return a failure message

endif

Note that LABEL runs in time $O(n^{s+2})$ whenever C-acceptable vertices can be
recognized in $O(n^s)$ steps for some constant $s$.

Let $C^4$ denote the class of all graphs on which LABEL succeeds. (This
class is well-defined: the success of LABEL is independent of the choice of $v$
in each iteration.)

THEOREM 6. Let $C$ be any t-bounded class of graphs such that members of
$C$ can be recognized in $O(n^{2t+2})$ steps, and C-acceptable vertices can be
recognized in $O(n^{2t+1})$ steps. Then any MWCP on any graph in $C^4$ can be solved in $O(n^{2t+3})$ steps.

Trivially, a vertex $v$ in a graph $F$ is TR-acceptable if and only if $F-v$ contains no disjoint sets $S_1, S_2$ of vertices such that both $F(S_1 \cup \{v\})$ and $F(S_2 \cup \{v\})$ are chordless cycles of length at least four. To put it differently, $v$ is not TR-acceptable if and only if $F-v$ contains vertex-disjoint paths $P_1, P_2$ such that each $P_i$ has at least two edges, and its terminal points $x_i, y_i$ are adjacent to $v$ in $F$. For each fixed choice of $x_1, y_1, x_2, y_2$, the existence of $P_1, P_2$ can be tested by efficient algorithms designed independently by Seymour (1980) and Shiloach (1980). In particular, Shiloach's algorithm runs in time $O(n^3)$; it follows that TR-acceptable vertices can be recognized in $O(n^7)$ steps.

**Corollary 6A.** Any MWCP on any graph in $TR^4$ can be solved in $O(n^9)$ steps.

Note that $TR^3 \not\subseteq TR^4$ and $TR^4 \not\subseteq TR^3$: the antihole with seven vertices belongs to $TR^3-TR^4$ and the graph shown in Fig. 1 belongs to $TR^4-TR^3$.

![Figure 1](image-url)
However, Corollary 6A holds with $TR^4$ replaced by a certain class $TR^5$ such that $TR^3 \cup TR^4 \subseteq TR^5 \subseteq TR^2$.

More generally, consider an arbitrary class $C$ of graphs. LABEL, given an arbitrary graph $G$, will produce some induced subgraph $F$ of $G$ and a labeling $v_{k+1}, v_{k+2}, \ldots, v_n$ of the vertices in $G-F$. It is an easy exercise to show that $F$ depends only on $G$ and $C$; we set $G \in C^5$ if and only if $F \in C^3$.

Trivially, a $C$-formative vertex partition of any $G$ in $C$ can be constructed by first taking an arbitrary labeling $v_1, v_2, \ldots, v_k$ of the vertices of $F$ and then applying GREEDY.

**THEOREM 7.** Let $C$ be any $t$-bounded class of graphs such that members of $C$ can be recognized in $O(n^2 t + 2)$ steps, and $C$-acceptable vertices can be recognized in $O(n^2 t + 1)$ steps. Then any MWCP on any graph in $C^5$ can be solved in $O(n^2 t + 3)$ steps.

**COROLLARY 7A.** Any MWCP on any graph in $TR^5$ can be solved in $O(n^9)$ steps.

Examples of graphs in $TR^5 - (TR^3 \cup TR^4)$ can be obtained by joining a graph in $TR^4$ to a graph in $TR^3$ by an appropriate set of edges. For instance, joining the vertices 1, 6 and 7 of the graph in Figure 1 by three edges to any three vertices of the antihole on seven vertices yields a graph in $TR^5$.

As we have observed, $C^5 \subseteq C^2$; since $G \in C^4$ if and only if $F$ has no vertices at all, we have $C^4 \subseteq C^5$. Finally, if $C$ is hereditary in the sense that every induced subgraph of every graph in $C$ is also in $C$, then $C^3$ is hereditary, and so $C^3 \subseteq C^5$.

Variations on the theme of $C^5$ abound. For instance, observe that all bipartite graphs and complements of all bipartite graphs belong to $TR^2$; in fact, a $TR$-formative vertex partition of each of these graphs can be found in.
O(n^2) steps. This observation suggests setting G ∈ TR^6 if and only if F ∈ TR^3 or F is bipartite or the complement of F is bipartite.

THEOREM 8. Any MWCP on any graph in TR^6 can be solved in O(n^9) steps.

Trivially, we have TR^5 ⊆ TR^6 ⊆ TR^2. Examples of graphs in TR^6 - TR^5 can be obtained by joining a graph in TR^4 to a bipartite graph not in TR^3 ∪ TR^4 by an appropriate set of edges. For instance, joining the vertices 1, 6, 7 of the graph shown in Fig. 1 by three edges to any three vertices of K_{5,5} - e (where e is an arbitrary edge of the complete bipartite graph K_{5,5}) yields a graph in TR^6 - TR^5. As examples of graphs in TR^2 - TR^6, we mention the antiholes with n vertices for n ≥ 9.

In closing this section, we recall that members of TR^4 are recognizable in O(n^9) steps. We know no polynomial-time algorithm for testing membership in TR^3 or TR^5 or TR^6, although for each of these classes there is a polynomial time algorithm that, given any graph G, solves the MWCP on G or shows that G does not belong to the class.

2. PROOFS

We begin with two well-known facts; their proofs are included for the sake of completeness.

FACT 1. There is an algorithm that, given any graph G, lists all the maximal cliques in G in O(n^2k^2) steps, with k standing for the number of items on the list.

PROOF. The following algorithm will do.
V-the vertex set of G; LIST=\{∅\}.

while V \neq ∅ do
  remove a vertex w from V
  N-the set of all neighbours of w that lie outside V
  t=0
  for all Q in LIST do
    P=Q \cap N, t=t+1, Q_t=P \cup \{w\}
    if P = Q then LIST=LIST - {Q} endif
  endfor
  for j = 1,2,...,t, do
    SMALL(j)=false
    for i = 1,2,...,j-1 do
      if Q_i \subseteq Q_j then SMALL(i)=true endif
      if Q_j \subseteq Q_i then SMALL(j)=true endif
    endfor
  endfor
  for j = 1,2,...,t do
    if SMALL(j) = false then LIST=LIST \cup \{Q_j\} endif
  endfor
endwhile.

Actually, the running time of an algorithm designed by Tsukiyama, Ide, Arioshi and Shirokawa (1977) to list all the maximal cliques in G is only \(O(n^3k)\), but \(O(n^2k^2)\) is good enough for our purpose.

FACT 2. There is an algorithm that, given any graph G along with two cliques in G whose union contains all the vertices of G, solves any MWCP on G in \(O(n^3)\) steps.
PROOF. The MWCP reduces to the problem of finding a minimum capacity cut in a network \( N \) constructed as follows. No generality is lost by assuming that the two cliques, \( Q_1 \) and \( Q_2 \), that cover all the vertices of \( G \) are disjoint. For nodes of \( N \), take all nodes of \( G \) and two extra nodes, \( s \) and \( t \); for arcs, take

\[ \begin{align*}
(*) & \text{ an arc } si \text{ of capacity equal to the weight of } i \text{ for each } i \in Q_1, \\
(*) & \text{ an arc } jt \text{ of capacity equal to the weight of } j \text{ for each } j \in Q_2, \\
(*) & \text{ an arc } ij \text{ of infinite capacity for each choice of } i \in Q_1 \text{ and } j \in Q_2 \text{ such that } i \text{ and } j \text{ are not adjacent in } G.
\end{align*} \]

A cut in a network with source \( s \) and sink \( t \) is any set \( S \) of nodes such that \( s \in S \) and \( t \not\in S \); the capacity of this cut is the sum of the capacities of all the arcs \( ij \) with \( i \in S, j \not\in S \). In our network, the capacity of a cut \( S \) is finite if and only if the set \( Q \) defined by

\[ Q = (Q_1 \cap S) \cup (Q_2 \setminus S) \]

is a clique of \( G \); if this is the case then the weight of \( Q \) and the capacity of \( S \) add up to the total weight of all the vertices of \( G \). Thus finding a clique of maximum weight in \( G \) amounts to finding a cut of minimum capacity in \( N \). The latter problem can be solved by a variety of efficient algorithms; in particular, an algorithm designed by Malhotra, Kumar, and Maheshwari (1978) runs in time \( O(n^3) \); for details of its implementation, see Chvátal (1983, pp. 380-386).

Our key notion is that of a clique basis, defined as a set of cliques \( Q_1, Q_2, \ldots, Q_k \) in a graph \( G \) such that each clique in \( G \) is a subset of some \( Q_i \cup Q_j \). Our key observation goes as follows.

FACT 3. There is an algorithm that, given any clique basis \( Q_1, Q_2, \ldots, Q_k \) in any graph \( G \), solves any MWCP on \( G \) in \( O(n^3k^2) \) steps.
(A proof is hardly required: to solve the MWCP on $G$, we only need to solve $k(k-1)/2$ problems on the subgraphs of $G$ induced by $Q_i \cup Q_j$.)

Our theorems will be proved in a permuted order.

**PROOF OF THEOREM 1.** By virtue of Facts 1 and 3, we only need to show that all the maximal cliques in the green graph constitute a clique basis in $G$. For this purpose, consider an arbitrary clique $Q$ in $G$ and let $F$ be the subgraph of the red graph induced by $Q$. By assumption, $F$ contains no triangle; since $Q$ is a clique in $G$ and since $C$ is tame, $F$ contains no odd hole. Thus $F$ is bipartite; to put it differently, $Q$ is covered by two cliques of the green graph.

**PROOF OF THEOREM 3.** We only need to observe that all the maximal cliques in the subgraphs of $G$ induced by $V_1$ and $V_2$ constitute a clique basis in $G$.

**PROOF OF THEOREM 4.** Lovász (1973) proved that the problem of recognizing bicolorable hypergraphs is NP-complete, and that it remains NP-complete even when the input is restricted to hypergraphs with all edges of size three. (A hypergraph is a collection of sets $E_1, E_2, \ldots, E_m$ called edges whose elements are called points; the hypergraph is called bicolorable if its points can be colored black and white in such a way that no edge is monochromatic.) It follows easily that recognizing bicolorable hypergraphs is an NP-complete problem even when the input is restricted to hypergraphs with all edges of size four. Given any such hypergraph $H$, we shall construct a graph $G$ such that $G \in \text{TR}^2$ if and only if $H$ is bicolorable.

First, let us construct a graph $F$ with distinguished vertices $x$ and $y$ such that $F \in \text{TR}^2$ and such that $x$ and $y$ must belong to distinct parts of every TR-formative partition of the vertex set of $F$. This is easy to do: take a $K_{3,3}$ (the complete bipartite graph with three vertices in each part),
add two nonadjacent vertices \( x \) and \( y \), and join each of these two by edges to all the vertices of the \( K_{3,3} \).

To construct \( G \), we create a copy of \( F \) for every pair \((E,p)\), where \( E \) is an edge of \( H \) and \( p \in E \); the two distinguished vertices of this \( F \) will be labeled \( x(E,p) \) and \( y(E,p) \). Then for each \( p \), we identify all the vertices labeled \( y(E,p) \), \( y(E',p) \), \( y(E'',p) \), ..., and label the resulting single vertex \( p^* \). Finally, for each edge \( E \) of \( H \), we enumerate the elements of \( E \) as \( a, b, c, d \) and create a chordless cycle of length four in \( G \) by joining each of \( x(E,a) \), \( x(E,b) \) to each of \( x(E,c) \), \( x(E,d) \).

To see that \( G \in \text{TR}^2 \) only if \( H \) is bicolorable, consider any \( \text{TR} \)-formative partition of the vertices of \( G \) and, referring to the two parts as "black" and "white", assign to each \( p \) in \( H \) the color of \( p^* \) in \( G \). We only need to show that no edge \( E \) of \( H \) is monochromatic. For this purpose, enumerate the elements of \( E \) as in the construction of \( G \) and note that the vertices \( x(E,a) \), \( x(E,b) \), \( x(E,c) \), \( x(E,d) \), inducing a chordless cycle of length four, cannot all have the same color. Since the color of each \( x(E,p) \) differs from that of \( p^* \), the desired conclusion follows.

To see that \( H \) is bicolorable only if \( G \in \text{TR}^2 \), consider any bicoloring of \( H \). Transfer the color of each \( p \) in \( H \) to \( p^* \) in \( G \), give each \( x(E,p) \) the color that differs from the color of \( p^* \) and, in each \( F \) used in the construction of \( G \), color one part of the \( K_{3,3} \) black and the other part white. Clearly, each of the two color classes induces a forest in \( G \).

PROOF OF THEOREM 2. Given any graph \( H \) we shall construct a graph \( G \) such that \( G \in \text{TR}^1 \) if and only if \( H \in \text{TR}^2 \).

First, let us construct a graph \( F \) with a distinguished edge \( xy \) such that \( F \in \text{TR}^1 \) and \( xy \) must be green in every \( \text{TR} \)-formative edge coloring of \( F \). This can be done by taking any graph \( F_0 \) of a chromatic number greater than six that
contains no triangle, adding two adjacent vertices $x$ and $y$, and joining each of these two vertices by edges to all the vertices of the $F_0$. (Trivially, $F \in \text{TR}^1$: color all the edges of $F_0$ red, and all the remaining edges of $F$ green. To see that $xy$ must be green in every TR-formative edge coloring of $F$, assume the contrary. Writing $z \in V_x$ if $zx$ is green, and $z \in V_y$ if $zy$ is green, observe that each vertex of $F_0$ belongs to $V_x \cup V_y$, and that each of the three graphs induced in $F_0$ by $V_x \cap V_y$, $V_x - V_y$ and $V_y - V_x$ is bipartite. This contradicts the fact that the chromatic number of $F_0$ exceeds six.)

To construct $G$, we take disjoint graphs $F_1, F_2, F_3, H$ such that each $F_i$ is a copy of $F$ with distinguished edge $x_i y_i$. Then we identify $y_1$ with $x_2$, identify $y_2$ with $x_3$, add edge $x_1 y_3$, and join each of the two vertices $x_1, y_3$ by edges to all the vertices of $H$.

To see that $G \in \text{TR}^1$ only if $H \in \text{TR}^2$, consider any TR-formative edge coloring of $G$. Since $x_1 y_1, x_2 y_2$ and $x_3 y_3$ are all green, $x_1 y_3$ must be red. For each vertex $v$ of $H$, write $v \in V_x$ if $vx_1$ is green, and write $v \in V_y$ if $vy_3$ is green. Clearly, each vertex of $H$ belongs to precisely one of these two sets (else $vx_1 y_3$ would be a red triangle or $vx_1 x_2 x_3 y_3$ would be a green chordless cycle) and each edge of $H$ with both endpoints in the same set is green. It follows that $V_x$ and $V_y$ form a TR-formative vertex partition of $H$.

A straightforward reversal of this argument shows that every TR-formative vertex partition of $H$ yields a TR-formative edge coloring of $G$; hence $H \in \text{TR}^2$ only if $G \in \text{TR}^1$.

Theorems 5-8 and their collaries require no proofs: they follow easily from Theorem 3 and the observations made in Section 1.
3. COMPARISONS WITH PREVIOUS RESULTS

The size of an instance of the maximum-weight clique problem is, roughly speaking, the amount of space required to record the data:

\[
\text{size} = n + m + \sum_{i=1}^{n} \log_{10} (1 + w_i)
\]

with \( n \) standing for the number of vertices, \( m \) for the number of edges, and \( w_i \) for the weight of the \( i \)-th vertex. An algorithm for solving the MWCP with input graphs restricted to some class \( \mathcal{C} \) is referred to as a polynomial-time algorithm if, for some constant \( t \), it solves any MWCP on any \( G \) in \( \mathcal{C} \) in \( O(\text{size}^t) \) steps. We shall now discuss classes \( \mathcal{C} \) for which such algorithms have been designed; for each of these classes \( \mathcal{C} \), we shall point out graphs that belong to \( \mathcal{C} \) but do not belong to \( \mathcal{C} \).

First, a graph is called perfect if, for each of its induced subgraphs \( F \), the chromatic number of \( F \) equals the largest size of a clique in \( F \). A polynomial-time algorithm for solving the MWCP on perfect graphs has been designed by Grotschel, Lovász, and Schrijver (1984a). Chordless cycles whose length is odd and at least five are not perfect, but do belong to \( \mathcal{C} \).

Second, Grotschel, Lovász and Schrijver (1984b) also designed a polynomial-time algorithm for solving the MWCP on complements of \( h \)-perfect graphs, defined as graphs for which the convex hull of the incidence vectors of stable sets is given by the clique inequalities, the odd cycle inequalities, and the nonnegativity conditions. The graph shown in Fig. 2 is not \( h \)-perfect, but its complement belongs to \( \mathcal{C} \).
Third, a graph is called **claw-free** if it contains no induced subgraph with vertices $x,y,z,w$, whose edges are precisely $xw,yw,zw$. Minty (1980) designed a polynomial-time algorithm for solving the MWCP on complements of claw-free graphs (the unweighted case was settled independently by Sbihi (1978)). Graphs consisting of two vertex-disjoint cliques with at least five vertices altogether are not complements of claw-free graphs, but they trivially belong to $\text{TR}^3 \cap \text{TR}^4$.

Fourth, Hsu, Ikura and Nemhauser (1981) designed a polynomial-time algorithm for solving the MWCP on graphs whose complements contain no odd cycle longer than an arbitrary but fixed constant $k$. Graphs with more than $k+1$ vertices and with no edges at all do not belong to this class, but they trivially belong to $\text{TR}^3 \cap \text{TR}^4$.

Finally, Fact 1 implies that for every $t$-bounded class $C$ there is an algorithm that solves any MWCP on any graph in $C$ in $O(n^{2t+2})$ steps. To see that no $t$-bounded class contains $\text{TR}^3 \cap \text{TR}^4$, consider the sequence of graphs:
G_1, G_2, G_3, ... such that G_k has vertices v_1, v_2, ..., v_{2k} and the only nonadjacent pairs are v_1v_2, v_3v_4, ..., v_{2k-1}v_{2k}. It is easy to see that G_k ∈ TR^3 ∩ TR^4 for all k. On the other hand, G_k has 2^k maximal cliques, and so no t-bounded class contains all G_k.

4. EXTENSIONS, LIMITATIONS, APPLICATIONS

The observation underlying our method is that any MWCP on a graph G can be solved quickly whenever the vertex set of G is covered by a small number of sets S_1, S_2, ..., S_N such that

(i) each clique in G is a subset of some S_i,

(ii) each S_i induces in G the complement of a bipartite graph.

(In Fact 3, we have N = \binom{k}{2} and each S_r is some Q_i ∪ Q_j.) The only reason for featuring complements of bipartite graphs in (ii) is that any MWCP on any of these graphs can be solved quickly; we could just as well use any other class C of graphs such that, for some constant t, some algorithm solves the MWCP on any G in C in O(size^t) steps. We shall refer to such classes C as t-solvable; their examples include the four classes discussed in the preceding section (perfect graphs, complements of h-perfect graphs, complements of claw-free graphs, and complements of graphs with no odd cycle longer than a constant), as well as the classes TR^3, ..., TR^6 introduced in Section 2. To generalize Fact 3, let us define a C-cover in G as any collection of subsets S_1, S_2, ..., S_k of the vertex set of G such that each clique in G is a subset of some S_i, and such that each S_i induces in G a member of C. The generalization goes as follows:

FACT 4. For every t-solvable class C there is an algorithm that, given any graph G along with a C-cover S_1, S_2, ..., S_k solves any MWCP on G in O(k size^t) steps. ||
Now let $f(G, C)$ stand for the smallest $k$ such that $G$ admits a $C$-cover $S_1, S_2, \ldots, S_k$. Clearly, Fact 4 provides a polynomial-time algorithm for solving every MWCP on every graph in some class $C^*$ only if, for some $t$-solvable class $C$ and for all $G$ in $C^*$,

$$f(G, C) \text{ does not exceed a fixed polynomial in } n. \quad (4.1)$$

We are going to show that, for every class $C$ that satisfies a certain technical assumption, a randomly chosen graph $G$ is extremely unlikely to satisfy (4.1). To make this claim precise, let us first clarify the meaning of "extremely unlikely". For this purpose, consider any property $P$ that a graph may or may not have, and let $P(n)$ equal the number of graphs with vertices $v_1, v_2, \ldots, v_n$ that have the property.

It is customary to say that almost all graphs have property $P$ if

$$\lim_{n \to \infty} \frac{P(n)}{2^{n(n-1)/2}} = 1 \quad (4.2)$$

(Observe that the denominator in (4.2) counts the number of graphs with vertices $v_1, v_2, \ldots, v_n$, and so the ratio in (4.2) equals the probability that a randomly chosen graph with these $n$ vertices has property $P$.)

Throughout the remainder of this section, we write $\log x$ for $\log_2 x$.

**THEOREM 9.** Let $C$ be any hereditary class of graphs other than the class of all graphs. Then for every positive $\epsilon$, almost all graphs $G$ have

$$f(G, C) > n \left( \frac{1}{2} - \epsilon \right) \log n. \quad ||$$

Our proof of Theorem 9 is based on two lemmas.

**LEMMA 1.** For every graph $F$ there is a constant $c$ such that almost all graphs $G$ have the following property: every induced subgraph of $G$ with at least $c \log n$ vertices contains an induced subgraph isomorphic to $F$. 
PROOF. Let \( k \) stand for the number of vertices of \( F \). Writing

\[
a = \exp \left( \left( \frac{1}{2} \right)^k \frac{k(k-1)}{2} \frac{1}{2k^2} \right),
\]

we shall prove the statement with \( c = 2/\log a \). To begin, let \( p(s) \) denote the probability that a randomly chosen graph with vertices \( w_1, w_2, \ldots, w_s \) has no induced subgraph isomorphic to \( F \). We only need to show that

\[
n \lim_{n \to \infty} \left( \sqrt[n]{n} p \right) (\sqrt[n]{c \log n}) = 0.
\]

For this purpose, let \( t(k,s) \) stand for the largest number \( t \) of sets \( Q_1, Q_2, \ldots, Q_t \) such that

- \( |Q_i| = k \) for all \( i \),
- \( |Q_i \cap Q_j| \leq 1 \) whenever \( i \neq j \), and
- \( \left| \bigcup_{i=1}^{t} Q_i \right| \leq s. \)

It is easy to see that

\[
p(s) \leq p(k) t(k,s) \leq \left( 1 - \left( \frac{1}{2} \right)^k \frac{k(k-1)}{2} \right) t(k,s) \leq \exp \left( - \left( \frac{1}{2} \right)^k \frac{k(k-1)}{2} t(k,s) \right).
\]

Erdos and Hanani (1963) have shown that

\[
s \lim_{s \to \infty} t(k,s) = \left( \frac{k}{2} \right) \left( \frac{2}{5} \right) = 1 \text{ for all } k;
\]

In particular, \( t(k,s) \geq s^2/2k(k-1) \geq s^2/2k^2 \) whenever \( s \geq s_0(k) \). Hence

\[
p(s) \leq a^{-s^2} \text{ whenever } s \geq s_o(k),
\]
and so

\[ \binom{n}{s} p(s) \leq (a^{-s})^s \text{ whenever } s \geq s_0(k). \]

Since \( a^{-c} \log n = n^{-2} \), the desired result follows.||

**Lemma 2.** For every positive \( \delta \), almost all graphs \( G \) have at least

\[ \frac{1}{2} - \delta \log n \]

cliques of size \( \log n \).||

**Proof.** As customary, we shall denote by \( P(A) \) the probability of event \( A \), and we shall let \( E(X) \) stand for the expected value of a random variable \( X \). We shall rely on the Chebyshev inequality, stating that

\[ P(X \leq E(X) - t) \leq \frac{E(X^2) - (E(X))^2}{E(X^2) - (E(X))^2 + t^2} \tag{4.3} \]

In addition, we shall use the fact that

\[ \sum_{i=0}^{k} \binom{k}{i} \binom{n-k}{k-i} t^i \leq (1 + (t-1) \frac{k}{n})^k \text{ whenever } t \geq 1 \tag{4.4} \]

(for an elementary proof, see Chvátal (1979)).

Now let \( n \) and \( k \) be fixed and let a random variable \( X \) count the number of cliques of size \( k \) in a randomly chosen graph with vertices \( v_1, v_2, \ldots, v_n \). Clearly,

\[ E(X) = \binom{n}{k} \left( \frac{1}{2} \right)^{\binom{k}{2}} \]

and
\[ E(X^2) = \binom{n}{k} \sum_{i=0}^{k} \binom{k}{i} \binom{n-k}{k-i} \frac{2^k}{i^2} - \binom{i}{2} \]

Note that

\[ \frac{E(X^2)}{(E(X))^2} = \frac{\sum_{i=0}^{k} \binom{k}{i} \binom{n-k}{k-i} \frac{2^k}{i^2}}{\binom{n}{k}} \binom{i}{2} \]

and so (4.4) with \( t = 2^{k/2} \) implies

\[ 1 \leq \frac{E(X^2)}{(E(X))^2} \leq \exp \frac{k^2 2^{k/2}}{n}. \]

Substituting into (4.3) we obtain

\[ P(X \leq \frac{1}{2} E(X)) \leq 4(\exp \frac{k^2 2^{k/2}}{n} - 1). \] (4.5)

In addition, note that

\[ E(X) \geq \left( \frac{n-k}{k 2^{k/2}} \right)^k. \]

In particular, if \( k = \lfloor \log n \rfloor \) then

\[ n^{1-\frac{1}{2} E(X)} \frac{1}{n^{(1/2-\delta) \log n}} = +\infty \text{ and } n^{1-\frac{1}{2} E(X)} \exp \frac{k^2 2^{k/2}}{n} = 1, \]

and so the desired result follows from (4.5).

PROOF OF THEOREM 9. By assumption, there is a graph \( F \) such that no graph in \( C \) contains an induced subgraph isomorphic to \( F \). We only need show that all graphs \( G \) with the two properties specified in Lemma 1 and Lemma 2 have
\[
\frac{1}{2} - \delta \log n - c
\]

For this purpose, consider an arbitrary C-cover \(S_1, S_2, \ldots, S_k\) in \(G\). W.l.o.g., we may assume that each \(S_i\) is minimal; then by Lemma 1, we have \(|S_i| \leq c \log n\) for all \(i\). Now Lemma 2 implies

\[
\left(\frac{1}{2} - \delta\right) \log n \leq \sum_{i=1}^{k} |S_i| \leq k n^c,
\]

which is the desired conclusion. \(\|

Theorem 8 shows that for a randomly chosen graph \(G\), Fact 4 is very unlikely to yield a polynomial-time algorithm for solving the MWCP on \(G\). On the other hand, the results of this paper can be used to devise improved heuristics or enumerative (non-polynomial-time) algorithms for solving the MWCP on an arbitrary graph. Let \(C\) be any t-solvable class for some constant \(t\), and suppose that for an arbitrary graph \(G\), a maximal induced subgraph \(G(S)\) of \(G\) that belongs to \(C\) can be generated in polynomial time. This is the case, for instance, with the classes \(TR^1\) and \(TR^2\) introduced in Section 2. If \(Q\) is a maximum-weight clique of \(G(S)\), then any clique of larger weight than \(Q\) must contain some vertex of \(V \setminus S\). Denoting by \(N(v)\) the neighbor set of vertex \(v\), one can branch by replacing \(G\) with the collection of induced subgraphs \(G(N(v_1)), G(N(v_2) \setminus \{v_1\}), \ldots, G(N(v_p) - [v_1, \ldots, v_{p-1}])\) where \([v_1, \ldots, v_p] = V \setminus S\). A branch and bound algorithm of this type, using as \(C\) the class of graphs whose chromatic number equals their maximum clique size, was proposed by Balas and Yu [1984] for the unweighted maximum clique problem. The algorithm was tested on randomly generated graphs with up to 400 vertices and 30,000 edges with considerably better results than earlier procedures based on straightforward
branch and bound. The classes of graphs introduced in this paper can be used in a similar fashion to obtain algorithms for the MWCP on general graphs.
REFERENCES


ERDÖS, P. (1959), "Graph theory and probability", Canad. J. Math. 11, 34-38


MINTY, G.J. (1980), "On maximal independent sets of vertices in claw-free graphs", J. Combinatorial Theory (B) 28, 284-304


SBIHI, N. (1978), These de 3ème cycle, Université de Grenoble. See also N. Sbihi, "Algorithme de recherche d'un stable de cardinalité maximum dans un graphe sans étoile", Discrete Math. 19 (1980), 53-76


ON THE MAXIMUM-WEIGHT CLIQUE PROBLEM

Egon Balas
Vašek Chvátal
Jaroslav Nešetřil

Graduate School of Industrial Administration
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

We introduce several new classes of graphs on which the maximum-weight clique problem is solvable in polynomial time. Their common feature, and the central idea of our algorithms, is that every clique of any of our graphs is contained in some member of a polynomial-sized collection of induced subgraphs that are complements of bipartite graphs. Our approach is quite general, and might conceivably yield many other classes of graphs along with corresponding polynomial time algorithms.