Graphs and Combinatorics 5, 339–349 (1989)

Graphs and Combinatorics

Optimizing Weakly Triangulated Graphs

Ryan Hayward[1]*, Chính Hoàng[1]*, and Frédéric Maffray[2]**

[1] Computer Science Department, Rutgers University, New Brunswick, NJ 08903, USA
[2] Rutgers Center for Operations Research, Rutgers University, New Brunswick, NJ 08903, USA

Abstract. A graph is weakly triangulated if neither the graph nor its complement contains a chordless cycle with five or more vertices as an induced subgraph. We use a new characterization of weakly triangulated graphs to solve certain optimization problems for these graphs. Specifically, an algorithm which runs in $O((n + e)n^3)$ time is presented which solves the maximum clique and minimum colouring problems for weakly triangulated graphs; performing the algorithm on the complement gives a solution to the maximum stable set and minimum clique covering problems. Also, an $O((n + e)n^4)$ time algorithm is presented which solves the weighted versions of these problems.

1. Introduction

Let C_k represent the chordless cycle with k vertices and P_k the chordless path with k vertices. Let $\bar{G}$ represent the complement of the graph G. A graph is *weakly triangulated* if it does not contain C_k or $\bar{C}_k$ as an induced subgraph, for any $k \geqq 5$. See [5] for an introduction to weakly triangulated graphs.

A *clique* of a graph is a subset K of the vertices, such that every two vertices in K are adjacent. An *independent set* of a graph, also called a *stable set*, is a subset S of the vertices, such that no two vertices in S are adjacent. A *colouring* of the vertices of a graph is a mapping of colours to the vertices of a graph, such that every two adjacent vertices receive different colours. Note that in a colouring of a graph, every set of vertices with the same colour is a stable set; thus a colouring can be thought of as a partition of the vertices of a graph into stable sets. A *clique covering* is a partition of the vertices of a graph into cliques.

In this paper we present polynomial time algorithms which solve the following problems: find a largest clique, a largest stable set, a minimum colouring, and a minimum clique covering of a weakly triangulated graph. We also present algorithms which solve the weighted versions of these problems (see Sect. 3).

* The author acknowledges the support of an N.S.E.R.C. Canada postgraduate scholarship.
** The author acknowledges the support of the U.S. Air Force Office of Scientific Research under grant number AFOSR 0271 to Rutgers University.

Claude Berge [1] defined a graph G to be *perfect* if, for every induced subgraph H of G, the chromatic number of H is equal to the size of a largest clique of H. (Weakly triangulated graphs are perfect; see [5].) For arbitrary graphs, the above optimization problems are *NP-complete*; see [3] and [7]. However, Grötschel, Lovász and Schrijver [4] have presented polynomial time algorithms, based on the ellipsoid method, which solve these problems for perfect graphs. The algorithms of Grötschel, Lovász and Schrijver are complex, and rely on deep properties of linear programming (see the last section of [4]). On the other hand, the algorithms presented in this paper are easily analyzed, and rely only on certain properties of weakly triangulated graphs, which we now describe.

2. Weakly Triangulated Graphs and Two-Pairs

A *two-pair* is a pair of (non-adjacent) vertices in a graph, such that every chordless path between the two vertices has exactly two edges.

The WT Two-Pair Theorem. *Every weakly triangulated graph which is not a clique has a two-pair.*

Proof. We prove a slightly stronger statement, namely, that all weakly triangulated graphs G other than cliques satisfy the following two properties:
(1) *if G has no clique cutset then each cutset of G contains a two-pair,*
(2) *G contains a two-pair.*

Arguing by induction on the number of vertices, we may assume that both (1) and (2) hold for all weakly triangulated graphs with fewer vertices than G. To prove (1) for G, consider any minimal cutset C of G. By assumption, C is not a clique. Define G_C as the subgraph of G induced by C. We shall distinguish between two cases.

Case 1. Suppose that $\bar{G}_C$ is disconnected. Let D be the set of vertices of some component $\bar{G}_C$ with at least two vertices (since C is not a clique, there must be such a set D). Note that every vertex of $C - D$ sees (is adjacent to) every vertex of D, and that D is a minimal cutset, not a clique, of $G - (C - D)$. Thus by inductive assumption, D contains a two-pair of $G - (C - D)$; obviously, this two-pair is a two-pair of G.

Case 2. Suppose that $\bar{G}_C$ is connected. Let $B_1, \ldots, B_t$ be the vertex sets of the components of $G - C$. Now (from Theorem 1 in [5]) it follows that in each component B_j, there is some vertex that sees every vertex of C.

Case 2.1. Suppose that $|B_j| = 1$ for all j. Then, by inductive assumption the graph G_C contains some two-pair $\{x, y\}$. Clearly $\{x, y\}$ is a two-pair of G.

Case 2.2. Suppose that $|B_j| \geqq 2$ for some j. Let z be any vertex of B_j that sees all of C; let E be the set of vertices of C that see some vertex of $B_j - z$. Observe that E is a cutset of $G - z$; let A be a subset of E so that A is a minimal cutset of $G - z$. We may assume that A is not empty, and not a clique (otherwise $A \cup \{z\}$ is a clique cutset of G, contradiction). Thus, by inductive assumption A contains a two-pair of $G - z$ which is clearly a two-pair of G.

To prove (2) for G, we may assume that G has a clique cutset C (otherwise the desired conclusion follows from (1)). Let $B_1, B_2, \ldots, B_t$ be the vertex sets of the components of $G - C$. If some $G - B_j$ is not a clique then by the induction hypothesis $G - B_j$ contains a two-pair; since every chordless path in G with both endpoints in $G - B_j$ is fully contained in $G - B_j$, this two-pair is also a two-pair in G. Hence we may assume that each $G - B_j$ is a clique. This implies that $t = 2$ and that $\{x, y\}$ is a two-pair whenever $x \in B_1$, $y \in B_2$. □

Note that for $k \geqq 5$, neither C_k nor $\bar{C}_k$ has a two-pair. Thus the above theorem implies this characterization: a graph is weakly triangulated if and only if every induced subgraph either is a clique or else has a two-pair.

An *even pair* is a pair of (non-adjacent) vertices of a graph, such that every chordless path which joins the two vertices has an even number of edges. Meyniel [8] defined a graph G to be *strict quasi-parity* if every induced subgraph H of G is a clique or has an even pair; a graph is *quasi-parity* if for every induced subgraph H of G, at least one of H or $\bar{H}$ is a clique or has an even pair. An immediate corollary of the *WT Two-Pair Theorem* is that weakly triangulated graphs are strict quasi-parity. However, weakly triangulated graphs can be recognized in polynomial time (e.g. see [6]), whereas it is not known whether this is true of quasi-parity graphs or strict quasi-parity graphs.

Similarly, the existence of a two-pair in a graph is a stronger condition than the existence of an even pair, and it is easy to check in polynomial time whether or not a pair of vertices is a two-pair (remove any common neighbours, and check whether there is a path between the original two vertices), whereas it is not known if determining the existence of an even pair can be done in polynomial time. In the next section we build upon the "find a two-pair" algorithm and construct polynomial time algorithms for solving the aforementioned optimization problems for weakly triangulated graphs.

3. The Algorithms

In this section algorithms are presented which solve the following problems for weakly triangulated graphs in polynomial time.

The Maximum Clique Problem. *Find a largest clique in a graph.*
The Maximum Stable Set Problem. *Find a largest stable set in a graph.*
The Minimum Colouring Problem. *Find a partition of the vertices into the smallest number of stable sets.*
The Minimum Clique Covering Problem. *Find a partition of the vertices into the smallest number of cliques.*

Algorithms are also presented which solve the weighted versions of these problems. In each of the following problems, assume that a graph G with vertices $v_1, \ldots, v_n$ and positive integers $w(v_1), \ldots, w(v_n)$ are given. These integers are referred to as *weights*.

The Maximum Weighted Clique Problem. *Find a clique K of G, such that the sum of the weights of the vertices of K is maximum, over all cliques of G.*
The Maximum Weighted Stable Set Problem. *Find a stable set S of G, such that the sum of the weights of the vertices of S is maximum, over all stable sets of G.*
The Minimum Weighted Colouring Problem. *Find stable sets $S_1, \ldots, S_t$ and integers $X(S_1), \ldots, X(S_t)$, such that*
(1) *for every vertex v_j, the sum of the integers $X(S_i)$ of all sets S_i such that $v_j \in S_i$ is at least $w(v_j)$, and such that*
(2) *the sum of all integers $X(S_1) + \cdots + X(S_t)$ is minimum, over all sets of integers that satisfy* (1).
The Minimum Weighted Clique Covering Problem. *Find cliques $K_1, \ldots, K_t$ and integers $X(K_1), \ldots, X(K_t)$, such that*
(1) *for every vertex v_j, the sum of the integers $X(K_i)$ of all sets K_i such that $v_j \in K_i$ is at least $w(v_j)$, and such that*
(2) *the sum of all integers $X(K_1) + \cdots + X(K_t)$ is minimum, over all sets of integers that satisfy* (1).

An algorithm which solves any of the weighted problems can be used to solve the unweighted version of the problem by assigning the weight "1" to all vertices. However, since our algorithms for the unweighted problems are more transparent and more efficient (in the sense of worst time complexity) than the algorithms for the weighted problems, we include both sets of algorithms.

Actually, we present only two algorithms. Algorithm OPT solves the maximum clique and minimum colouring problem for weakly triangulated graphs; Algorithm W-OPT solves the weighted versions of these problems. Since the complement of a weakly triangulated graph is weakly triangulated, Algorithms OPT and W-OPT can also be used to solve the unweighted and weighted versions respectively of the maximum stable set and minimum clique covering problems: to find a largest stable set of a graph G, find a largest clique of $\bar{G}$; to find a minimum clique covering of a graph G, find a minimum colouring of $\bar{G}$.

Our algorithms rely on the fact that every weakly triangulated graph is either a clique or else has a two-pair (see the previous section). The aforementioned optimization problems are easily solved for graphs which are cliques. Given a weakly triangulated graph other than a clique, our algorithms repeatedly find a two-pair, each time transforming the graph in question into a smaller weakly triangulated graph by "identifying" the two-pair. (We will define this term shortly.) Eventually the original graph is transformed into a clique; the optimization problem is solved for the clique, and the two-pair identification process is reversed, transforming the solution of the optimization problem for the clique to the solution of the optimization problem for the original graph.

3.1. The Unweighted Case

Let $G(xy \to z)$ be the graph obtained by replacing vertices x and y of G with a vertex z, such that z sees exactly those vertices of $G - \{x, y\}$ that see at least one of $\{x, y\}$. The *identification* of x and y in G is the process of replacing G with $G(xy \to z)$.

In the following algorithm, we specify a colouring by a function f_G that assigns some integer from 1 to t to each vertex, such that adjacent vertices are assigned different integers. Assume that $V(G) = \{v_1, v_2, \ldots, v_n\}$ is the set of vertices of G.

Algorithm OPT(G)
Input: a weakly triangulated graph G.
Output: a largest clique K_G and a minimum colouring f_G.
Step 1. Look for a two-pair $\{x, y\}$ of G.
If G has no two-pair, then
(a) $K_G \leftarrow V(G)$,
(b) for $i = 1$ to n do $f_G(v_i) \leftarrow i$, and
(c) **STOP**.
Step 2. $J \leftarrow G(x, y \rightarrow z)$.
Step 3. $K_J, f_J \leftarrow$ **OPT**(J).
Step 4a. If $z \notin K_J$, then $K_G \leftarrow K_J$, else ($z \in K_J$ and ...)
if x sees all of $K_J - \{z\}$ then $K_G \leftarrow K_J - \{z\} + \{x\}$,
else $K_G \leftarrow K_J - \{z\} + \{y\}$.
Step 4b. $f_G(x) \leftarrow f_G(y) \leftarrow f_J(z)$;
for each $v_i \in J - \{x, y\}$ do
$f_G(v_i) \leftarrow f_J(v_i)$. □

To prove the correctness of Algorithm OPT, we need to establish several properties concerning the identification of a two-pair in a weakly triangulated graph. One such property is described in the following lemma.

The Identification Lemma. *Let G be a weakly triangulated graph with a two-pair $\{x, y\}$. Then $G(xy \rightarrow z)$ is weakly triangulated.*

Proof. Let $H = G(xy \rightarrow z)$. We prove that if H is not weakly triangulated, then neither is G. Assume that H is not weakly triangulated. Then there is some subset C of the vertices of H, such that the subgraph H_C of H induced by C is either C_k or $\bar{C}_k$, with $k \geqq 5$. If $z \notin C$, then clearly G is not weakly triangulated. Thus we may assume that $z \in C$.

Case 1. H_C is a chordless cycle $c_1 \ldots c_k$ with $k \geqq 5$.

Assume without loss of generality that $z = c_1$. Then $c_2, \ldots, c_k$ is a chordless path in G. Since z sees c_2, c_k, and none of $c_3, \ldots, c_{k-1}$, at least one of $\{x, y\}$ sees c_2, and similarly c_k, and neither x nor y sees any of $\{c_3, \ldots, c_{k-1}\}$. Now observe that at least one of $\{x, y\}$ must see both of $\{c_2, c_k\}$. (Suppose not; assume w.l.o.g. that x sees c_2 but not c_k and that y sees c_k but not c_2. Then $(x, c_2, \ldots, c_k, y)$ is a chordless path with at least six vertices, contradicting the assumption that $\{x, y\}$ is a two-pair.) Thus assume w.l.o.g. that x sees both of $\{c_2, c_k\}$. Then $\{x, c_2, \ldots, c_k\}$ induces a C_k in G, G is not weakly triangulated, and the theorem holds in this case.

Case 2. $\bar{H}_C$ is a chordless cycle $c_1 \ldots c_k$ with $k \geqq 5$.

Assume without loss of generality that $z = c_1$. Thus $c_2 \ldots c_k$ is a $\bar{P}_{k-1}$ in G, and
(i) c_2 sees neither x nor y and c_k sees neither x nor y, and
(ii) every vertex in $\{c_3, \ldots, c_k\}$ sees at least one of $\{x, y\}$.

Now observe that
(iii) x or y sees both c_3 and c_4.
(Assume the contrary. By (ii) either x or y sees c_3; assume w.l.o.g. that x sees c_3. Since (iii) does not hold, x does not see c_4; thus by (ii) y sees c_4, and since (iii) does not hold, y does not see c_3. But then (x, c_3, c_k, c_4, y) is a P_5, contradicting the fact that $\{x, y\}$ is a two-pair in G.)

Assume w.l.o.g. that x sees both c_3 and c_4; let m be the smallest index greater than four such that x does not see c_m. Then $xc_2 \ldots c_m$ is a $\bar{C}_k$, with $k \geqq 5$. G is not weakly triangulated, and the theorem holds in this case. □

Another result that will be used in proving the correctness of Algorithm OPT is that two-pair identification does not change the clique size. This follows from a lemma due to Meyniel.

The Clique Size Lemma (Meyniel [8]). *If vertices x and y of a graph G are not joined by any chordless path with three edges, then $\omega(G(xy \to z)) = \omega(G)$.*

The Clique Size Corollary. *If $\{x, y\}$ is a two-pair of the weakly triangulated graph G, then $\omega(G(xy \to z)) = \omega(G)$.*

The Correctness Theorem. *Algorithm OPT finds a largest clique and a minimum colouring of G.*

Proof. Throughout the proof we let $|f_G|$ and $|f_J|$ denote the number of colours of f_G and f_J respectively. Since the clique size of a graph is never greater than the chromatic number, to prove the theorem it suffices to show that K_G is a clique, that f_G is a colouring, and that $|K_G| = |f_G|$. The proof is by induction on the number of calls of OPT. (Since identification decreases the number of vertices by one, OPT is called at most n times; thus the algorithm terminates.) If OPT is called only once, then the algorithm terminates at Step 1. By the WT Two-Pair Theorem, $K_G = V(G)$ is a clique, f_G is a colouring with $n = |K_G|$ colours, and the theorem holds.

Suppose then that OPT is called more than once; thus the algorithm terminates with Step 4b. Since (by the *Identification Lemma*) J is weakly triangulated, by the inductive hypothesis we may assume that K_J and f_J are respectively a clique and a colouring of J, such that $|K_J| = |f_J|$. If $z \notin K_J$, then $K_G = K_J$, and $|K_G| = |K_J|$. If $z \in K_J$, then either x or y must see all vertices of $K_J - z$. (Suppose not. Then x misses some $v_i \in K_J$; however, y sees v_i, else z would miss v_i. Similarly, y misses some $v_j \in K_J$ that sees x. But then xv_jv_iy is a chordless path, contradicting the assumption that $\{x, y\}$ is a two-pair.) Thus $|K_G| \geqq |K_J|$. Since K_J is a largest clique of J, the Identification Lemma implies that $|K_G| = |K_J|$.

Since no pair of adjacent vertices a, b of J satisfy $f_J(a) = f_J(b)$, no pair of adjacent vertices a, b of $G - \{x, y\}$ satisfy $f_G(a) = f_G(b)$. Finally, let c be a vertex of G that sees at least one of $\{x, y\}$; then c sees z in J, and so

$$f_G(c) = f_J(c) \neq f_J(z) = f_G(x) = f_G(y).$$

Thus no pair of adjacent vertices u, v of G satisfy $f_G(u) = f_G(v)$, and f_G is a colouring. Note that $|f_G| = |f_J|$. Thus $|K_G| = |K_J| = |f_J| = |f_G|$, and the theorem is proved. □

A corollary of the Correctness Theorem is that $\omega(G) = \chi(G)$ if G is weakly triangulated. Thus (since every induced subgraph of a weakly triangulated graph is weakly triangulated) the Correctness Theorem yields another proof that weakly triangulated graphs are perfect.

We now analyze the complexity of Algorithm OPT(G). Let e be the number of edges of G, and n the number of vertices. Note that a pair of non-adjacent vertices x and y in a graph G is a two-pair if and only if there is no path from x to y in $G - N$, where N is the set of all vertices of G that see both x and y. Determining whether or not two vertices are in the same component of a graph can be done in time $O(n + e)$. Thus determining whether or not a pair of vertices is a two-pair can be done in time $O(n + e)$, and Step 1 can be done in time $O((n + e)n^2)$. Step 2 can be done in time $O(n)$, as can Steps 4a and 4b. Since Step 3 is executed at most $n - 1$ times, the worst-case complexity of Algorithm OPT is $O((n + e)n^3)$.

Note that Algorithm OPT can be used to solve the Maximum Stable Set and Minimum Clique Covering problems of a graph G by taking as input the complement $\bar{G}$; in this case, the complexity will be $O((n + \bar{e})n^3)$, where $\bar{e}$ is the number of edges of $\bar{G}$.

3.2. The Weighted Case

In this section we present polynomial time algorithms that solve the weighted versions of the maximum clique, maximum stable set, minimum colouring and minimum clique covering problems for weakly triangulated graphs.

One way to solve the weighted clique problem for a graph G is to replace every vertex u with vertices v and w, such that v sees w, and such that u, v, w see exactly on the resulting graph. However, this transformation is inefficient if the weights are large. Our solution is more direct.

Define $G(u \to vw)$ to be the graph obtained from the graph G by replacing the vertex u *with vertices* v and w, such that v sees w, and such that u, v, w see exactly the same vertices of $G - u$. This process is referred to as *duplication*.

We now define an operation that combines identification and duplication. Define $G(xy \to za)$ to be the graph $H(xb \to z)$, where $H = G(y \to ab)$. We refer to the process of replacing G with $G(xy \to za)$ as *quasi-identification*.

Quasi-identification is represented in Fig. 1. Note that $G(xy \to za)$ is the graph obtained from G by replacing x, y with z, a respectively, such that z sees a, z sees exactly those vertices of $G - \{x, y\}$ that see at least one of $\{x, y\}$, and a sees exactly those vertices of $G - \{x, y\}$ that see y.

In the following algorithm, the weighted colouring f_G consists of stable sets $S_{G_1}, S_{G_2}, \ldots, S_{G_r}$ and associated positive integers $X(S_{G_1}), X(S_{G_2}), \ldots, X(S_{G_r})$.

Algorithm W-OPT(G).

Input: a weakly triangulated graph G.

Output: a max. weighted clique K_G and a min. weighted colouring f_G.

Step 1. Look for a two-pair $\{x, y\}$ of G.
If G has no two-pair then
(a) $K_G \leftarrow V(G)$,
(b) for $i \leftarrow 1$ to n do

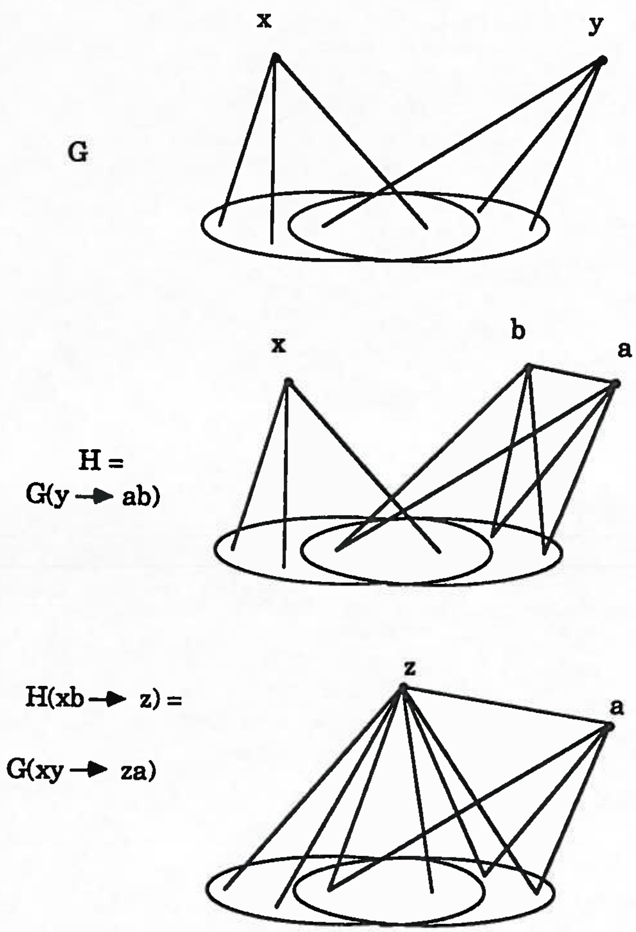

Fig. 1. Quasi-identification

$S_{G_i} \leftarrow \{v_i\}$,
$X(S_{G_i}) \leftarrow w(v_i)$

(c) **STOP**.

Step 2. Assume that $w(x) \leqq w(y)$.
If $w(x) = w(y)$ then
$J \leftarrow G(xy \rightarrow z)$,
$w(z) \leftarrow w(x)$;
else $\{\ldots$ thus $w(x) < w(y) \ldots\}$
$J \leftarrow G(xy \rightarrow za)$,
$w(z) \leftarrow w(x)$,
$w(a) \leftarrow w(y) - w(x)$.

Step 3. $K_J, f_J \leftarrow$ **W-OPT**(J).

Step 4a. If $z \notin K_J$ then $K_G \leftarrow K_J$ else $(z \in K_J$ and$\ldots)$
if y sees all of $K_J - \{a, z\}$ then $K_G \leftarrow K_J - \{a, z\} + y$
else $(\ldots x$ sees all of $K_J - \{a, z\} \ldots)$ $K_G \leftarrow K_J - \{a, z\} + x$.

Step 4b. For each set S_{J_i} of f_J do
(i) if $z \in S_{J_i}$ then $S_{G_i} \leftarrow S_{J_i} - z + \{x, y\}$, else
if $a \in S_{J_i}$ then $S_{G_i} \leftarrow S_{J_i} - a + y$, else
$S_{G_i} \leftarrow S_{J_i}$,
(ii) $X(S_{G_i}) \leftarrow X(S_{J_i})$. □

The proof of correctness of Algorithm W-OPT parallels the proof of correctness of Algorithm OPT. We first show that quasi-identification of a two-pair of a weakly triangulated graph yields a weakly triangulated graph.

The Quasi-Identification Lemma. *Let G be a weakly triangulated graph with a two-pair* $\{x, y\}$. *Then* $G(xy \to za)$ *is weakly triangulated.*

Proof. $G(xy \to za) = H(xb \to z)$, where $H = G(y \to ab)$. It is easy to check that H is weakly triangulated and that $\{x, b\}$ is a two-pair of H. Now the result follows from the Identification Lemma. □

Next we prove that the process of quasi-identification, together with the re-weighting of the new vertices as described in Algorithm W-OPT, does not change the weighted clique number of G. Let $\Omega(G)$ represent the weighted clique number of G (i.e. the weight of a maximum weighted clique of G).

The Weighted Clique Number Lemma. *Let G be a weighted weakly triangulated graph with a two-pair* $\{x, y\}$ *such that* $w(x) \leqq w(y)$. *Let* $F = G(xy \to za)$, *and let* $w(z) = w(x)$ *and* $w(a) = w(y) - w(x)$. *Then* $\Omega(G) = \Omega(F)$.

Proof. $F = G(xy \to za) = H(xb \to z)$, where $H = G(y \to ab)$. Let $w(b) = w(x)$; clearly $\Omega(H) = \Omega(G)$. To prove the lemma we need only show that $\Omega(F) = \Omega(H)$.

Let K_H be a clique of H of maximum weight. Since x, b are non-adjacent, K_H contains at most one of these two vertices. If K_H contains neither x nor b, then K_H is a clique of F. If K_H contains x, then $K_H - x + z$ is a clique of F with the same weight as K_H; if K_H contains b, then $K_H - b + z$ is a clique of F with the same weight as K_H. Thus $\Omega(F) \geqq \Omega(H)$.

Now let K_F be a clique of F of maximum weight. If $z \notin K_F$ then K_F is a clique of H; if $z \in K_F$ then either $K_F - z + x$ or $K_F - z + b$ is a clique of H, and both have the same weight as K_F. Thus $\Omega(H) \geqq \Omega(F)$. □

Essentially, the proof of the correctness of W-OPT can be derived from the proof of the correctness of OPT, observing that each step of W-OPT is equivalent to a polynomial number of steps of OPT. For those who prefer a more detailed argument, we include the following theorem and proof.

The Weighted Correctness Theorem. *Algorithm* W-OPT *solves the Maximum Weighted Clique Problem and the Minimum Weighted Colouring Problem for a weakly triangulated graph G.*

Proof. Let K_G and f_G be as described in Algorithm W-OPT. It is easy to check that K_G is a clique, and that S_{G_i} is a stable set, for all i. Let $|K_G| = \sum_{v \in K_G} w(v)$ and let

$|f_G| = \sum_i X_{G_i}$. We wish to show that f_G satisfies property (1) of the definition of the Minimum Weight Colouring Problem, and that $|K_G| = |f_G|$. Note that if K is any clique of a weighted graph, and if f is any colouring that satisfies (1), then $|K| \leqq |f|$; thus the equality $|K_G| = |f_G|$ implies that both K_G and f_G are optimal.

We first show that (1) holds for f_G. Argue by induction on the number of times Step 1 is executed in W-OPT(G). If Step 1 is executed only once, then $X(S_{G_i}) = v_i$ for all $i = 1, \ldots, n$, and (1) holds.

Suppose then that Step 1 is executed at least twice. Thus the algorithm terminates with Step 4. Assume by induction that (1) holds for the colouring f_J of J. Recall that in Step 4b,

the vertex z is replaced (in every set S_{J_i} of f_J that contains z) with the pair of vertices x, y, and, if $w(x) < w(y)$,

the vertex a is replaced (in every set S_{J_i} of f_J that contains a) with the vertex y. In the case where $w(x) = w(y)$, we have $w(z) = w(x) = w(y)$, and so

$$w(x) = w(z) = \sum_{S_{J_i} \supseteq z} X(S_{J_i}) = \sum_{S_{G_i} \supseteq x} X(S_{G_i}),$$

$$w(y) = w(z) = \sum_{S_{J_i} \supseteq z} X(S_{J_i}) = \sum_{S_{G_i} \supseteq y} X(S_{G_i}).$$

In the case where $w(x) < w(y)$, we have $w(x) = w(z)$ and $w(y) = w(a) + w(z)$, and so

$$w(x) = w(z) = \sum_{S_{J_i} \supseteq z} X(S_{J_i}) = \sum_{S_{G_i} \supseteq x} X(S_{G_i}),$$

$$w(y) = w(z) + w(a) = \sum_{S_{J_i} \supseteq z} X(S_{J_i}) + \sum_{S_{J_i} \supseteq a} X(S_{J_i}) = \sum_{S_{G_i} \supseteq y} X(S_{G_i}).$$

Thus property (1) holds for f_G.

Now we wish to show that $|K_G| = |f_G|$. Argue by induction on the number of executions of Step 1; the result clearly holds if Step 1 is executed exactly once. Assume then that Step 1 is executed more than once; thus the algorithm terminates with Step 4. By the induction hypothesis, $|K_J| = |f_J|$.

Now an argument similar to that used in the Correctness Theorem establishes that $|K_G| = |K_J|$; thus to finish the proof, we need only show that $|f_G| = |f_J|$. But this is obviously the case, because there is a one-to-one correspondence between the stable sets of f_G and f_J, namely S_{G_i} corresponds to S_{J_i}, and $X(S_{G_i}) = X(S_{J_i})$ for all i. □

We now analyze the complexity of Algorithm W-OPT(G). Let e be the number of edges of G, and n the number of vertices. As in Algorithm OPT(G), Step 1 can be done in time $O((n + e)n^2)$, and Steps 2, 4a and 4b can be done in time $O(n)$. Now consider Step 3. The graph J is either $G(xy \to z)$ or $G(xy \to za)$. In the former case J has one vertex fewer than G; in the latter case, J has at least one edge more than G (z sees every vertex of $G - \{x, y\}$ that x sees, a sees every vertex of $G - \{x, y\}$ that y sees, and z sees a whereas x misses y). Thus Step 3 is executed at most $n - 1 + \binom{n}{2} - e$ times, and the worst-case complexity of Algorithm W-OPT is $O((n + e)n^4)$ arithmetic operations. Since the number of arithmetic operations is bounded by a polynomial in n and e, Algorithm W-OPT is (as is Algorithm OPT) strongly polynomial.

References

1. Berge, C.: Färbung von Graphen deren sämtliche bzw. ungerade Kreise starr sind (Zusammenfassung). Wiss. Z. Martin-Luther-Univ. Halle-Winterberg, Math.-Natur. Reihe, 114 (1961)
2. Berge, C. Chvátal, V.: *Topics on perfect graphs*, Ann. Discrete Math. **21**, (1984)
3. Cook, S.A.: The complexity of theorem-proving procedure. In: Proceedings of the Third Annual ACM Symposium on Theory of Computing, Assoc. Comp. Mach., New York, 151–158 (1971)
4. Grötschel, M., Lovász, L. Schrijver, A.: Polynomial algorithms for perfect graphs. In: [2, pp. 325–356]
5. Hayward, R.: Weakly triangulated graphs. J. Comb. Theory (B) **39**, 200–208 (1985)
6. Hayward, R.: Two classes of perfect graphs. Doctoral thesis, School of Computer Science, McGill University (1987)
7. Karp, R.: Reducibility among combinatorial problems. In: Complexity of Computer Computations (by R. Miller and J. Thatcher) pp. 85–103 New York: Plenum Press (1972)
8. Meyniel, H.: A new property of critical imperfect graphs and some consequences. Europ. J. Comb. **8**, 313–316 (1987)

Received: April 9, 1987

Graphs and Combinatorics 6, 33–35 (1990)

Graphs and Combinatorics

Erratum

Optimizing Weakly Triangulated Graphs

[Graphs and Combinatorics 5, 339–349 (1989)]

Ryan Hayward[1], Chính T. Hoàng[1] and Frédéric Maffray[2]

[1] Institüt für Diskrete Mathematik, Universität Bonn, Nassestrasse 2, D-5300 Bonn 1, Federal Republic of Germany

[2] RUTCOR, Rutgers Center for Operations Research, Rutgers University, New Brunswick NJ 08903, USA

Due to an oversight on the part of the authors, the proof given in [1] for *The WT Two-Pair Theorem* is incomplete, and should be replaced with the following proof.

Recall that a *two-pair* is a pair of non-adjacent vertices in a graph, such that every chordless path between the two vertices has exactly two edges.

The WT Two-Pair Theorem *Let G be any weakly triangulated graph. Then either G is a clique or it contains a two-pair. Moreover, if C is any minimal cutset of G, then either C is a clique or C contains a two-pair of G.*

Proof. We first make two observations.

Observation 1. Let X be a set of vertices of G and let $\{y, z\}$ be a two-pair of $G - X$ such that every vertex of X is adjacent to both y and z. Then $\{y, z\}$ is a two-pair of G.

Observation 2. Let F be a clique of a graph G, and let B^* be the union of some connected components of $G - F$. Then any two-pair $\{x, y\}$ of $G - B^*$ is a two-pair of G.

We prove the Theorem by induction on the number of vertices of G. We may assume that G is not a clique. If G is disconnected, then we obtain a two-pair by taking two vertices lying in two distinct components of G. (A graph is disconnected if and only if the only minimal cutset is the empty set; we consider the empty set as a clique-cutset.) We may thus assume that G is connected. Let C be a minimal cutset of G, and let $B_1, \ldots, B_p$ be the components of $G - C$. Define $G[C]$ as the subgraph of G induced by C. We shall distinguish between two cases.

Case 1. C is a clique of G.

If there is a component B_j of $G - C$ such that $G - B_j$ is not a clique, then by the induction hypothesis the graph $G - B_j$ has a two-pair, which is also a two-pair of G by Observation 2 (where $F = C$ and $B^* = B_j$). Else, we must have that $p = 2$ and

$C \cup B_j$ induces a clique for $j = 1$ and 2. Then $\{x_1, x_2\}$ is a two-pair for any $x_1 \in B_1$ and $x_2 \in B_2$.

Case 2. $\bar{G}[C]$ is disconnected.

Let C^* be the set of vertices of some component of $\bar{G}[C]$ with at least two vertices (since C is not a clique, there must be such a set C^*). Note that every vertex of $C - C^*$ is a neighbor of every vertex of C^*, and that C^* is a minimal cutset, and not a clique, of $G - (C - C^*)$. Thus by inductive assumption, C^* contains a two-pair of $G - (C - C^*)$; this two-pair is also two-pair of G by Observation 1 (where $X = C - C^*$).

Case 3. $\bar{G}[C]$ is connected.

From Hayward's Theorem [2] it follows that in each component B_j of $G - C$, there is some vertex that is a neighbor of every vertex of C. If each B_j consists of a single vertex, then by the induction hypothesis the subgraph $G[C]$ contains a two-pair, which is also a two-pair of G by Observation 1 (where $X = B_1 \cup \cdots \cup B_p$). We may then assume without loss of generality that B_1 has at least two vertices. Let x be any vertex of B_1 that is a neighbor of all of C. We shall distinguish among three subcases.

Subcase 3.1. $G - x$ is disconnected.

Remarking that $C \cup B_2 \cup \cdots \cup B_p$ is contained in one single component of $G - x$, we define B^* to be the union of all the other components of $G - x$ (thus $B^* \neq \phi$ and $B^* \subset B_1 - x$), and $B_0 = B_1 - B^*$. Clearly $G[B_0]$ is connected, for otherwise any component of $G[B_0]$ not containing x would be a connected component of $G - C$, contradicting the definition of B_1. Note that C is a minimal cutset of the graph $G - B^*$, whose components are $B_0, B_2, \ldots, B_p$. By the inductive hypothesis C contains a two-pair of $G - B^*$, which is also a two-pair of G by Observation 2 (where B^* is as defined above and $F = \{x\}$).

Subcase 3.2. $G - x$ is connected and C is a minimal cutset of $G - x$.

By the induction hypothesis, C contains a two-pair of $G - x$, which is a two-pair of G by Observation 1 (where $X = \{x\}$).

Subcase 3.3. $G - x$ is connected and C is not a minimal cutset of $G - x$.

Let C' be a minimal cutset, contained in C, of $G - x$. Note that C' is not empty because $G - x$ is connected, and that $C'' = C - C'$ is not empty because C is not a minimal cutset of $G - x$. If C' is not a clique then, by the induction hypothesis, C' contain a two-pair of $G - x$, which is also a two-pair of G by Observation 1 (where $X = \{x\}$). Now we may assume that C' induces a clique.

Since C is a minimal cutset of G, each vertex of C'' must have at least one neighbor in each B_j. Therefore $B_2 \cup \cdots \cup B_p \cup C''$ is included in one single component of $(G - x) - C'$. Let B^* be another component of $(G - x) - C'$. Then we have $B^* \subset B_1 - x$. Furthermore, a vertex a of B^* cannot be adjacent to any vertex b of $C'' \cup (B_1 - x - B^*)$, because a and b are in different components of $(G - x) - C'$. It follows that $G[B_1 - B^*]$ is connected, for otherwise any component of $G[B_1 - B^*]$ not containing x would form a connected component of $G - C$, contradicting the definition of B_1. Thus C is a minimal cutset of $G - B^*$, the components of $(G - x) -$

C being $B_1 - B^*, B_2, \ldots, B_p$, since each vertex of C is a neighbor of the vertex x of $B_1 - B^*$. By the induction hypothesis, C contains a two-pair of $G - B^*$, which is also a two-pair of G by Observation 2 (where $F = C' \cup \{x\}$ and B^* is as defined above). This completes the proof. □

The above proof is essentially that of [3], with all instances of "even pair" replaced with "two-pair".

References

1. Hayward, R., Hoàng C.T., Maffray, F.: Optimizing weakly triangulated graphs. Graphs and Combinatorics **5**, 339–349 (1989)
2. Hayward, R.: Weakly Triangulated Graphs. J. Comb. Theory (B) **39**, 200–208 (1985)
3. Hoàng, C.T., Maffray, F.: Weakly triangulated graphs are strict quasi-parity graphs. RUTCOR Research Rpt., Rutgers University 6–86 (1986)

Received: January 25, 1990