Discrete Applied Mathematics 27 (1990) 85-99
North-Holland

COMPLETELY SEPARABLE GRAPHS*

Peter L. HAMMER and Frédéric MAFFRAY**

RUTCOR, Rutgers Center for Operations Research, Hill Center, Busch Campus, Rutgers University, New Brunswick, NJ 08903, USA

Received 30 May 1989

We define a property of Boolean functions called separability, and specialize it for a class of functions naturally associated with graphs. "Completely separable graphs" are then derived and characterized in particular by the existence of two crossing chords in any cycle of length at least five. This implies that completely separable graphs are perfect. We present linear time algorithms for the recognition and for the usual optimization problems (maximum weighted stable set and maximum weighted clique).

1. Introduction

Many problems that arise in combinatorial optimization can be expressed as:

$$\begin{aligned} &\text{Maximize} && w\cdot x, \\ &\text{subject to} && f(x)=0, \end{aligned} \tag{1}$$

where f is a given Boolean function of the n-dimensional vector x. For example, the weighted stability number of a graph, in which the vertices $x_1,\dots,x_n$ are assigned weights $w_1,\dots,w_n$, is given by any solution of problem (1), when f is a certain Boolean function associated with the graph. Similarly, the maximum weight matching problem can be formulated as an instance of (1).

In the general case, problem (1) is known to be NP-complete. One idea to help solving such a problem is to try to reduce it to smaller (and easier) instances.

Definition 1.1. We will say that a Boolean function f on n variables $x_1,\dots,x_n$ is *separable* if we can write

$$f(x)=\alpha(A)\vee\alpha'(A)\cdot\beta'(B)\vee\beta(B),$$

where $\alpha(A)$ and $\alpha'(A)$ are Boolean functions of the variables of a set $A\subseteq\{x_1,\dots,x_n\}$, and $\beta(B)$ and $\beta'(B)$ are Boolean functions of the variables of a set $B\subseteq\{x_1,\dots,x_n\}$, with $|A|\ge 2$, $|B|\ge 2$, and $A\cap B=\emptyset$.

* A preliminary version of this paper was presented at the International Colloquium on Graph Theory and Combinatorics, Marseille-Luminy, France, June 1986.

** The authors acknowledge the support of the US Air Force Office of Scientific Research under grant number AFOSR 85-0271 and 89-0066 to Rutgers University and of the NSF grant ECS 85 03212.

Definition 1.2. A Boolean function is said to be *positive* if it can be expressed in a normal disjunctive form in which no complemented variable appears.

An *implicant* $\pi(x)$ of a Boolean function $f(x)$ is any term such that $\pi(x)=1 \Rightarrow f(x)=1$. A *prime implicant* of $f(x)$ is any implicant of $f(x)$ from which the deletion of any literal appearing in it produces a nonimplicant.

It is well known that any Boolean function is equal to the union of all its prime implicants.

We will write $p(A)$, $q(B)$ etc., to mean that the Boolean function p depends only on the variables of A, that q depends only on the variables of B, etc. Also we will write $p(A)\equiv 0$, or $p(A)\equiv 1$, when we mean that $p(A)$ is constantly equal to 0, or to 1.

Theorem 1.3. *Let f be a positive, separable Boolean function with the notation of Definition* 1.2. *Then there exists a quadruple $(\alpha^*(A), \alpha^{**}(A), \beta^*(B), \beta^{**}(B))$ of positive Boolean functions such that*

$$f(x)=\alpha^*(A)\vee\alpha^{**}(A)\cdot\beta^{**}(B)\vee\beta^{**}(B).$$

Proof. Let $\alpha^*(A)$ be the union of all prime implicants $p(A)$ of $f(x)$. Let $\beta^*(B)$ be the union of all prime implicants $q(B)$ of $f(x)$. Let $\alpha^{**}(A)$ be the union of all terms $p(A)\not\equiv 1$ such that $p(A)\cdot r(B)$ is a prime implicant of $f(x)$ for some $r(B)\not\equiv 1$. Let $\beta^{**}(B)$ be the union of all terms $q(B)\not\equiv 1$ such that $s(A)\cdot q(B)$ is a prime implicant of $f(x)$ for some $s(A)\not\equiv 1$. Since every prime implicant of $f(x)$ is positive, all the terms $p(A)$, $q(B)$, $r(B)$, $s(A)$ mentioned above are positive and thus all the functions $\alpha^*(A)$, $\alpha^{**}(A)$, $\beta^*(B)$, $\beta^{**}(B)$ are positive.

Clearly every implicant of the function $f(x)$ is an implicant of the function $\alpha^*(A)\vee\alpha^{**}(A)\cdot\beta^{**}(B)\vee\beta^*(B)$. So, in order to prove that these two functions are equal, we just need to show that any prime implicant $\pi(x)$ of $\alpha^*(A)\vee\alpha^{**}(A)\cdot\beta^{**}(B)\vee\beta^*(B)$ is a prime implicant of $f(x)$. This is true if $\pi(x)$ is a prime implicant of $\alpha^*(A)$ or $\beta^*(B)$ by the definition of these functions. If $\pi(x)$ is not of this type, then necessarily $\pi(x)=p(A)\cdot q(B)$, where $p(A)\not\equiv 1$ and is such that $p(A)\cdot r(B)$ is a prime implicant of $f(x)$ for some $r(B)\not\equiv 1$, and $q(B)\not\equiv 1$ and is such that $s(A)\cdot q(B)$ is a prime implicant of $f(x)$ for some $s(A)\not\equiv 1$.

Note that if $p(A)\cdot r(B)$ is a prime implicant of $f(x)$ with $p(A)\not\equiv 1$ and $r(B)\not\equiv 1$, then in fact $p(A)\cdot r(B)$ is a prime implicant of $\alpha'(A)\cdot\beta'(B)$, which in turn implies that $p(A)$ is a prime implicant of $\alpha'(A)$ and $r(B)$ is a prime implicant of $\beta'(B)$, because $A\cap B=\emptyset$. Similarly $q(B)$ is a prime implicant of $\beta'(B)$ and $s(A)$ is a prime implicant of $\alpha'(A)$. Now $\pi(x)=1$ implies $p(A)=1$ and $q(B)=1$, which imply $\alpha'(A)=1$ and $\beta'(B)=1$. Consequently $\alpha'(A)\cdot\beta'(B)=1$, and finally $f(x)=1$. □

2. Graphic separable functions and their graphs

Definition 2.1. A Boolean function is said to be *quadratic* if it can be expressed in a normal disjunctive form as a union of terms each of which is the product of at most two literals.

A quadratic and positive Boolean function is called a *graphic* function.

This latest adjective appears because of the following correspondence between graphic functions and graphs: Let f be a graphic function of the variables $x_1, \dots, x_n$. We can build an (undirected) graph G having vertex set $V = \{x_1, \dots, x_n\}$ such that $x_i x_j$ is an edge of G whenever $x_i x_j$ is a term of f. In particular, a linear term $x_i (= x_i x_i)$ of f corresponds to a loop in G. Conversely, one can build a Boolean function f from any graph G by using the same procedure. Note however that f will be in normal disjunctive form if and only if every loop of G is incident to an isolated vertex. In the following, we will always assume that G has no loops and no multiple edges.

Definition 2.2. A graph is *separable* if the corresponding graphic Boolean function is separable, or if it has at most three vertices. A graph is *completely separable* if every induced subgraph is separable.

Before we go further, some terminology need to be introduced.

For any vertex x of G, $N(x)$ denotes the set of all neighbors of x. A vertex having exactly one neighbor is called a *pendant* vertex. A *homogeneous set* of a graph G is a set of vertices A such that every vertex in $V(G) - A$ is adjacent to either all or none of the vertices of A. A *proper homogeneous set* is a homogeneous set A such that $2 \le |A| \le n-2$, where n is the number of vertices in G. Two vertices are called *twins* if they form a homogeneous set of size 2. *True* twins are adjacent, *false* twins are not. A *hinge* of G is a partition of $V(G)$ in four sets A'', A', B', B'', with $|A' \cup A''| \ge 2$ and $|B' \cup B''| \ge 2$, and such that there is no edge between A'' and B', between A' and B'', and between A'' and B'', and all possible edges exist between A' and B'. Note that if A is a proper homogeneous set of G, then $(\emptyset, A, N(A), V(G) - A - N(A))$ is a hinge. A cycle of G with at least five vertices will be called a *long* cycle. A *chord* of a cycle is any edge xy joining two nonconsecutive vertices of the cycle. Two chords ac and bd are *crossing* if the four vertices a, b, c, d lie in this order on the cycle. The chordless path on four vertices is denoted by P_4.

Theorem 2.3. *A graph G with at least four vertices is separable if and only if it has a hinge.*

The concept of separation is a special case of the decomposition considered by Cunningham and Edmonds [8]. It is also studied by Bixby [3]. Using their terminology, our completely separable graphs correspond to those which decompose into indecomposable graphs of size at most three.

Proof of Theorem 2.3. First suppose that G is separable and has at least four vertices. By Definition 2.2, the graphic Boolean function f associated with G is separable, which means that we can write

$$f(x)=\alpha(A)\vee\alpha'(A)\cdot\beta'(B)\vee\beta(B)$$

with the notation of Definition 1.1. Also, by Theorem 1.3, we can assume that α, α', β and β' are positive functions. We shall distinguish between two cases.

Case 1: *One of the functions α' and β', say α', is a constant.* If α' is the constant zero, we have $f(x)=\alpha(A)\vee\beta(B)$. This implies that in G there is no edge between the vertices of A and the vertices of B, since f contains no term with an element of A and an element of B. Therefore A is a homogeneous set, and $2\le|A|\le n-2$ holds by Definition 1.1. It follows from an earlier comment that G has a hinge.

If α' is the constant one, we have $f(x)=\alpha(A)\vee\beta'(B)\vee\beta(B)$. Here in fact the terms of $\beta'(B)$ might as well be included in $\beta(B)$, and we are essentially in the same situation as when α' is the constant zero.

Case 2: *None of the functions α' and β' is a constant.* Since f is quadratic, it follows that both α' and β' must be linear (and by Theorem 1.3 positive) functions, i.e., unions of uncomplemented variables.

Let A' (respectively B') be the set of variables that appear in $\alpha'(A)$ (respectively in $\beta'(B)$), and $A''=A-A'$ (respectively $B''=B-B'$). Then $\alpha'(A)=\alpha'(A')$ and $\beta'(B)=\beta'(B')$. Clearly now (A'',A',B',B'') forms a hinge.

We now show the converse part of the equivalence.

Suppose that G has a hinge (A'',A',B',B''). Let $A=A'\cup A''$ and $B=B'\cup B''$. Let $\alpha(A)$ (respectively $\beta(B)$) be the graphic function associated with the subgraph of G induced by A (respectively by B). Let $\alpha'(A)$ (respectively $\beta'(B)$) be the union of all uncomplemented variables a' (respectively b') such that the vertex a' is an element of A' (respectively b' is an element of B'). Then one can easily see that $f(x)=\alpha(A)\vee\alpha'(A)\cdot\beta'(B)\vee\beta(B)$ is the graphic function associated with G, and that it is separable, with the same notation as in Definition 1.1. □

Remark. If G is disconnected, it is separable. Indeed, consider a connected component A of G of minimal size. Then A is a proper homogeneous set of G, unless A consists of just one isolated vertex A of G. In that case, let B be another vertex of G, let C be its neighborhood and $D=V(G)-(\{a\}\cup\{b\}\cup C)$. Then either $(\{\mathrm{A}\},\{B\},C,D)$ forms a hinge, or C is empty (and thus $\{A,B\}$ is a proper homogeneous set), or D is empty (and thus C is a proper homogeneous set).

This theorem can be used to determine a collection of graphs which are not separable. Indeed the graphs shown in Fig. 1 (long chordless cycle, "house", "gem" and "domino") clearly contain no proper homogeneous set and no hinge. Furthermore they are minimal among nonseparable graphs. The following theorem shows that there is no other minimal forbidden subgraph, and characterizes all completely separable graphs.

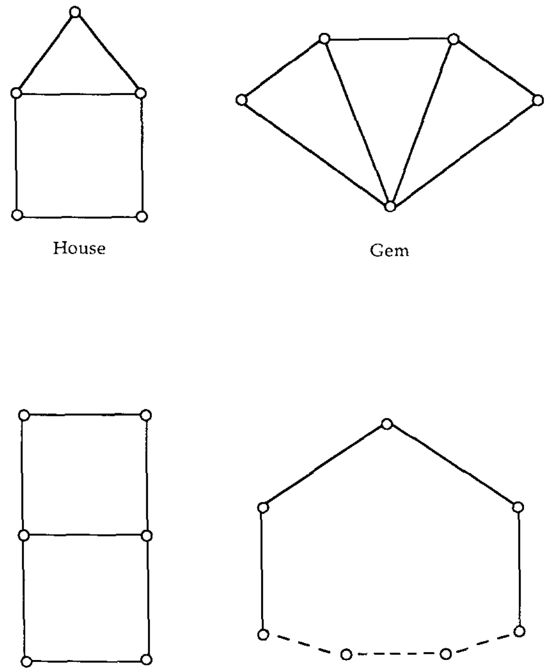

Fig. 1. Minimal noncompletely separable graphs.

Theorem 2.4. *The following five properties are equivalent*:

(1) *G is completely separable*;
(2) *G contains none of the configurations shown in Fig.* 1;
(3) *every long cycle of G has two crossing chords*;
(4) *every induced subgraph of G has a pair of twins or a pendant vertex*;
(5) *given any two vertices u and v of G, all chordless paths from u to v have the same length.*

Proof. The equivalence of properties (3) and (5) is stated in [12] where metric aspects of graphs with property (5) are studied.

The implication (1) $\Rightarrow$ (2) results from Theorem 2.3 as explained above.

The implication (2) $\Rightarrow$ (3) is shown by induction on the length k of a long cycle C of G. If $k=5$, then C must have two crossing chords otherwise its vertices would induce a pentagon or a house or a gem. If $k=6$, either the vertices of C induce a hexagon or a domino, which are forbidden, or C has two nontriangular chords and they necessarily cross each other, or it contains a cycle of length 5 and thus two crossing chords by the induction hypothesis. If $k \geq 7$, the cycle C must have a chord since G contains no long chordless cycle. This chord divides C in two shorter cycles, one of which has length at least 5. By the induction hypothesis this subcycle of C has two crossing chords, which in turn are crossing chords of C.

To prove (3) ⇒ (4), let G satisfy (3). It suffices to prove that G satisfies (4) since (3) is transmitted to all induced subgraphs. We note that (4) holds trivially if G is a disjoint union of cliques. We may thus assume that some component H of G is not a clique. Let Q be a minimal cutset of H and $R_1, \dots, R_m$ be the components of $H-Q$. Suppose that $|Q| \geq 2$; we show that Q is a homogeneous set. If not, there are two vertices p, q of Q and a vertex r of $V(H)-Q$ which is adjacent to p and not to q. Let r be in R_1. Since Q is a minimal cutset of H, vertex q has a neighbor s in R_1. Note that r and s are connected by a path in R_1. We choose s so that this path P_1 is as short as possible. Similarly p has a neighbor t in R_2 and q has a neighbor u in R_2. We choose t and u so that they are connected by a short as possible path P_2 in R_2 (maybe $t=u$). The vertices s, q, u, t, p, r and the paths P_1 and P_2 form a cycle C of length at least 5. The only possible chords of C join p to q or to some vertices of P_1. Therefore C has no crossing chords, a contradiction. Now if x is any vertex in R_1 which is adjacent to Q, it must be adjacent to all vertices of Q and thus Q is P_4-free. We know that a nontrivial P_4-free graph has a pair of twins (see [4, Lemma 5]). They will also be twins of G because Q is homogeneous.

Now suppose that every minimal cutset Q of H has $|Q|=1$. Let R be a terminal block of H, i.e., a maximal 2-connected subgraph of H that contains just one cut-vertex, say x, of H. We can observe that:

If $|R|=2$, the element in $R-x$ is a pendant vertex of G.

If $|R| \geq 3$ *and* $R-x \subseteq N(x)$, the set $R-x$ must induce a P_4-free subgraph otherwise G has a gem, which is a cycle without two crossing chords. So R contains a pair of twins [4], and clearly they are also twins in G.

If $R-N(x) \neq \emptyset$, $N(x) \cap R$ is a cutset of H and so it contains a minimal cutset of size 1. But then R is not 2-connected, a contradiction.

To prove (4) ⇒ (1), let H be any induced subgraph of G with at least four vertices. If H has a pair of twins, they form a proper homogeneous set of H. Otherwise, H has a pendant vertex x whose neighbor in H we call y. Clearly, $(\{x\}, \{y\}, N(y) \cap V(H) - \{x\}, V(H) - (N(y) \cup \{y\}))$ is a hinge of H. Therefore every induced subgraph of G has a hinge, and the conclusion follows from Theorem 2.3. □

After the completion of this manuscript, we became aware of [1, 9] in which some equivalences of Theorem 2.4 have been found independently.

What property (4) actually means is the following: Given any completely separable graph G on n vertices, there is an indexing $1, 2, \dots, n$ of its vertices and a sequence $(s_2, \dots, s_n)$ of "words" such that, for $2 \leq i \leq n$, the word s_i is respectively iPj, or iFj, or iTj, for some integer $j<i$, with the meaning that the subgraph of G induced by $\{1, 2, \dots, i\}$ is obtained from the subgraph induced by $\{1, 2, \dots, i-1\}$ by making vertex i respectively a pendant vertex to j, or a false twin or a true twin of j. The letter P, F or T will be called the *type* of vertex i, and j will be called the

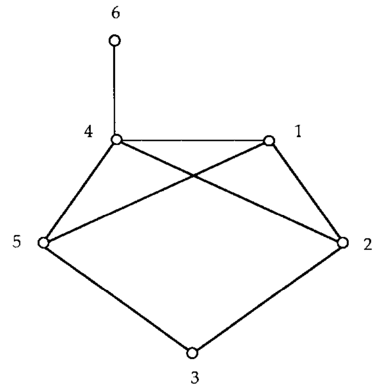

Fig. 2.

relative of i. This will be called a *pruning sequence* of G. For example $(2P1, 3P2, 4T1, 5F2, 6P4)$ is a pruning sequence of the graph represented on Fig. 2. Conversely, if G has a pruning sequence, then it cannot contain any forbidden configuration H shown in Fig. 1. This can be proved by induction on the number of vertices: If n is pendant in G, then n does not belong to H. If n is a twin of i, then i and n do not both belong to H. In any case, either H or $H \cup \{i\} - \{n\}$ is a forbidden subgraph of $G - \{n\}$, a contradiction to the induction hypothesis.

Corollary 2.5. *Trees and P_4-free graphs are completely separable.*

Proof. A tree can be iteratively built by adding a pendant vertex, so it satisfies property (4) of Theorem 2.4. A P_4-free graph cannot contain any of the forbidden subgraphs given in property (2). □

Among P_4-free graphs is the class of *threshold* graphs [6], which can be defined by the absence of any induced P_4 or C_4 or complement of C_4. It is easy to see that threshold graphs are exactly those completely separable graphs that have a pruning sequence in which every s_i is either $iP1$ or $iF1$ or $iT1$ (vertex 1 must have maximum degree), and all the F's come first (they correspond to the isolated vertices of the graph).

We recall from Berge [2] that a graph is *perfect* if the vertices of every induced subgraph H can be colored (in such a way that no two adjacent vertices receive the same color) with a number of colors equal to the maximal size of a clique of H. Two classes of perfect graphs have a special interes for us: *Brittle* graphs, as defined by Chvátal and studied in [10], are such that every induced subgraph H possesses a vertex which either is not the endpoint of any P_4 or is not the midpoint of any P_4 of H. *Parity* graphs [4, 13] are such that given any two vertices u and v of a parity graph, all chordless paths from u to v have the same length parity.

Proposition 2.6. *Completely separable graphs are parity graphs, and they are brittle.*

Proof. The fact that all completely separable graphs are parity is a trivial consequence of their satisfying property (5). Furthermore, it is proved in [11] that all graphs which do not contain a house, a domino or a long chordless cycle are brittle. □

The class inclusions stated in the proposition are strict. Indeed the domino is a parity graph and a brittle graph, but it is not completely separable.

Corollary 2.7. *Completely separable graphs are perfect.*

In fact, brittle graphs are *perfectly orderable*: There is a *perfect order* on the vertex set, i.e., a linear order $<$ such that for any P_4 on vertices a, b, c, d with edges ab, bc, cd, the inequalities $a<b$ and $d<c$ do not both hold. Chvátal [5] introduced perfectly orderable graphs and showed that they are perfect. It can be seen without difficulty that if G is a completely separable graph, the indexing of the vertices corresponding to a pruning sequence of G is a perfect order for G. Now if a perfectly orderable graph is given together with a perfect order on its vertices, it is possible to find an optimal coloration of the graph, just by assigning to each vertex the first color that has not been assigned to the neighbors which preceed it in the order. This procedure can be implemented to run in linear time in the number of edges of the graph, but of course the perfect ordering must have been determined beforehand. The search for a pruning sequence is the point of our next section.

3. Structural and algorithmic aspects

In [9] an $O(nm)$ algorithm is presented that decomposes any graph into its minimal nonseparable subgraphs. This algorithm may be used to find whether a graph is completely separable, just by testing if all the minimal subgraphs so obtained have less than four vertices. Here we will present an $O(n+m)$ algorithm that not only recognizes if a given graph is completely separable but also, if it is, outputs a pruning sequence for it. It hinges on the fact that the $O(n+m)$ algorithm presented in [7], which recognizes P_4-free graphs, can be used to find a pruning sequence for a given P_4-free graph.

Suppose that G is a completely separable graph and a is a vertex of G. Let L_i be the set of vertices at distance i from a, and p be the smallest integer such that L_{p+1} is empty. If x is in L_i and y is in L_j with $i \neq j$, we denote by $P(x,y)$ a path (if any) that goes from x to y and meets each of $N_i, N_{i+1}, \dots, N_j$ exactly once.

Any vertex u different from x that lies on any path $P(a,x)$ is an *ancestor* of x, and every vertex is a *descendant* of its ancestors. Two vertices are *tied* if they have a common descendant. For any x in L_i, let $N'(x) = N(x) \cap L_{i-1}$ and $d'(x) = |N'(x)|$.

Fact 3.1. *Two adjacent vertices of L_i have the same neighbors in L_{i-1}. Any connected component of L_p is a homogeneous set of G.*

Proof. This fact actually holds for any parity graph and is given as [4, Lemma 7]. The second sentence is a trivial consequence of the first one and the definition of p. □

Fact 3.2. *Two tied vertices of L_i have the same neighbors in L_{i-1}.*

Proof. Let x and y be tied vertices of L_i and let z be a closest common descendant of them. Suppose that some vertex u of L_{i-1} is adjacent to x and not to y. Let $v \in N'(y)$. Let b be a closest common ancestor of u and v. By walking along $P(b,u)$, the edge ux, $P(x,z)$, $P(z,y)$, the edge yv and $P(v,b)$ we define a cycle of length at least 6. By the definition of b and z, the only possible chords of this cycle are uv and xy and they are not crossing, a contradiction. □

Fact 3.3. *If two vertices x and y of L_i are such that $N'(x) \cap N'(y) \neq \emptyset$, then either $N'(x) \subseteq N'(y)$ or $N'(y) \subseteq N'(x)$.*

Proof. Suppose Fact 3.3 does not hold: There exist a vertex w in $N'(x) \cap N'(y)$, a vertex u in $N'(x) - N'(y)$ and a vertex v in $N'(y) - N'(x)$. Note that u and w are tied (by x) and that v and w are tied (by y). So, by Fact 3.2, the vertices u, v and w have a common neighbor t in L_{i-2}. It is easy to see that the subgraph of G induced by $\{t,u,v,w,x,y\}$ contains one of the forbidden configurations of Fig. 1, a contradiction. □

A family of subsets of a set such that any two elements of the family are either disjoint or comparable (by inclusion) is called *arboreal.*

We define an order $\leq_i$ between the vertices of L_i by $x \leq_i y \Leftrightarrow N'(x) \subseteq N'(y)$.

Fact 3.4. *If x is a minimal element of $(L_i, \leq_i)$, then $N'(x)$ is a homogeneous set of G.*

Proof. Suppose that a vertex y lying outside $N'(x)$ is adjacent to a vertex u of $N'(x)$ and not adjacent to another vertex v of $N'(x)$. If y is in L_i, then, by the definition of x, Fact 3.3 is contradicted. If y is in L_{i-2}, then Fact 3.2 is contradicted. So y must be in $L_{i-1} - N'(x)$. Since u and v are tied (by x) and u and y are adjacent, the vertices u, v and y must have a common neighbor t in L_{i-2}. But now $\{t,u,v,x,y\}$ induces a forbidden subgraph, a contradiction. □

Note that x is a minimal element of $(L_i, \leq_i)$ if and only if $d'(x) = \min\{d'(z); z \in L_i\}$.

Let $\approx_i$ be the relation defined between vertices of L_i by $x \approx_i y \Leftrightarrow x$ and y are in the same connected component of L_i or x and y are tied. Let $\approx_a$ be defined on $V(G)$ by $x \approx_a y \Leftrightarrow x \approx_i y$ for some i.

Fact 3.5. *The relation $\approx_a$ is an equivalence.*

Proof. Clearly, it suffices to prove that for any three vertices x, y, z in any level L_i, the following two assertions hold:

If x is adjacent to y and y is tied with z, then x is adjacent to or tied with z;

If x is tied with y and y is tied with z, then x is tied with z.

Suppose that the first assertion does not hold. Let u be a closest common descendant of y and z. By Facts 3.1 and 3.2, x, y and z have a common neighbor t in L_{i-1}. Vertex u must be in L_{i+1} since otherwise $P(u,z)$, zt, ty and $P(y,u)$ form a cycle of length at least 6 that does not have two crossing chords. But then $\{x,y,z,t,u\}$ is a forbidden induced subgraph of G, a contradiction.

If the second assertion does not hold, let u be a common descendant of x and y and v be a common descendant of y and z. Again, x, y and z must have a common neighbor t in L_{i-1}. As before, u and v must be in L_{i+1}. Note that x is not adjacent to v and u is not adjacent to z. Then $\{x,y,z,t,u,v\}$ is or contains a forbidden induced subgraph of G, a contradiction. □

Now the homogeneous sets mentioned in Facts 3.1 and 3.4 must be P_4-free, otherwise G contains a gem. We know that a nontrivial P_4-free graph has a pair of twins. Clearly, twins of a homogeneous set are twins of the whole graph. Therefore Facts 3.1 and 3.4 tell us where we will find a pair of twins of G: in any nontrivial connected component of L_p, or (if L_p is an independent set) in $N'(x)$, for a minimal element x of $(L_p, \leq_p)$; otherwise, this x is a pendant vertex of G.

We can now make explicit the structural aspects of our graphs.

Theorem 3.6. *Let a be a vertex of a connected, completely separable graph G and $L_1, \dots, L_p$ be the distance levels from a. Let $R_1, \dots, R_r$ be the equivalence classes of the relation $\approx_a$, and let $S_j = N(R_j) \cap L_{i-1}$, where i is such that $R_j \subseteq L_i$. Then:*

(1) *The graph obtained from G by shrinking each R_j into one vertex is a tree rooted at a;*

(2) *each R_j induces a P_4-free subgraph;*

(3) *for each j, the family $\{S_k : S_k \subseteq R_j\}$ is an arboreal family of homogeneous subsets of R_j.*

Conversely, if there exist two partitions $(R_1, \dots, R_r)$ and $(S_1, \dots, S_r)$ of $V(G)$ such that any S_k is a subset of some R_j, that all vertices of R_j are adjacent to all vertices of S_j and that (1), (2) and (3) are satisfied, then G is a completely separable graph.

Proof. Suppose that G is completely separable. Then (1) must hold because any vertex of G ties all its ancestors in a given level and therefore they are all shrunk together into one single vertex.

By Facts 3.1 and 3.2, all vertices of a given R_j have the same neighbors (and at least one) in the preceding level. Thus (2) holds otherwise G contains a gem.

(3) is just a reformulation of Facts 3.3 and 3.4.

Conversely, the properties (1), (2) and (3) clearly suffice to imply Facts 3.1 and 3.4 which, as explained after Fact 3.5, imply the existence of a pair of twins or a pendant vertex. $\square$

We will now exploit the Facts 3.1–3.5 to obtain an efficient recognition algorithm.

Algorithm PRUNE(G).

Input. A graph G with n vertices.

Output. A pruning sequence $(s_2, \ldots, s_n)$ for G if it is completely separable or else a subgraph of G that is isomorphic to one of the forbidden configurations in Fig. 1.

begin
  $W := V(G)$.
  while $W \neq \emptyset$ **do**
    Let $a \in W$. Put a in a stack. Build the distance levels $L_1, \ldots, L_p$ with respect to a.
    for $j = p, p-1, \ldots, 2$ **do**
      Round 1. Find the connected components of L_j.
      Round 2. For each component A of L_j with $|A| \geq 2$, call REDUCE(A).
      Round 3. (L_j is now a stable set.) Order the vertices of L_j by increasing degree $d'(x)$. Delete those with $d'(x) = 1$ (they are pendant vertices of $G(W)$).
      Round 4. For each vertex x of L_j taken by increasing degree $d'(x)$, call REDUCE($N'(x)$) (now $d'(x) = 1$), and delete x.
      (L_j is now empy.)
    endfor
    $W := W - a$.
  endwhile
  (Now the stack contains one vertex from each of the c connected components of G.)
  Let one vertex of the stack have index 1, and let the other ones have respectively indices $2, \ldots, c$ and all have type F and relative 1.
end

Subroutine REDUCE(A): This subroutine checks that the set A is a homogeneous set of G and that it induces a P_4-free subgraph. In this case it deletes one by one the twins of the set A until there is only one remaining vertex.

Step a. To check that A is homogeneous, we pick a vertex x of A and, for every vertex z of $A-x$, compare the list $N'(z)$ with the list $N'(x)$.

Step b. To check that the subgraph $G(A)$ induced by A is P_4-free, we use the algorithm in [7]. If $G(A)$ is P_4-free, this algorithm represents it by a unique rooted tree T such that

(i) the leaves of the tree correspond to the vertices of $G(A)$;
(ii) every internal node is labelled either 0 or 1;
(iii) every internal node (except maybe the root) has at least two children;
(iv) every node of the tree is labelled differently from its father;
(v) two vertices of $G(A)$ are adjacent if and only if they correspond to two leaves whose closest common ancestor is an internal node labelled 1.

Step c. It is easy to see that two vertices of $G(A)$ are twins if and only if they correspond to two leaves of T having a common father. When we have T, we can find the list of all the fathers of the leaves. We can also build the sublist D of those fathers that have at least two leaves. Since any nontrivial P_4-free graph has a pair of twins, we know that D is not empy. We take an element δ of D and delete from T (and from A and from W) all but one of its children: they correspond to twins of $G(A)$ and so of $G(W)$. We remove δ from T and append its remaining child to the father of δ; if this father becomes the father of two leaves, we insert it in D. The new tree *is* the representative of the subgraph corresponding to the undeleted nodes. We continue this procedure until D becomes empty, which necessarily means that A has been reduced to one single vertex.

If a call to REDUCE fails, we have a certain subset of $V(G)$ which is either not homogeneous or not P_4-free, and Fact 3.1 or Fact 3.4 enable us to find an induced forbidden configuration of G.

Note: Whenever we delete a vertex from W, except for the vertices of the stack, we know that it is a pendant vertex or a twin in $G(W)$. So we assign it the highest index available (initialized at n), and we note its type and the name of its relative.

Complexity analysis. (For any subgraph A of G, let $n(A)$ be the number of its vertices and $m(A)$ the number of its internal edges.) We may assume that G is given by the ordered adjacency list of each vertex.

At the beginning of each iteration of the "while" loop, we pick a vertex a and build the distance levels by a breadth-first search that runs in $O(m(H))$, where H is the connected component of G that contains a. We can also build in $O(m(H))$ the ordered lists $N'(x)$ for all vertices x of H.

During the execution of the subroutine REDUCE(A), Step a is done in time $n(A)+2\cdot\sum_{z\in A} d'(z)$ by comparing the ordered adjacency lists of the elements of A; Step b is done in $O(n(A)+m(A))$; Step c works in $O(n(T))$ and properties (i) and (ii) imply that $n(T)\le 2n(A)$. Therefore the subroutine REDUCE(A) works in time $K\cdot(n(A)+\sum_{z\in A} d'(z))$ for some constant K.

During the jth iteration of the "for" loop, Rounds 1, 2 and 3 are executed in time $O(n(L_j)+\sum_{z\in L_j} d'(z))$. At each iteration of Round 4, some subset of L_{j-1} is reduced to one vertex and this vertex remains in L_{j-1}. Thus its degree might be counted many times. However, we know that whenever a vertex remains, one of its twins (having the same degree) is pruned off the graph. This implies that the total running time of Round 4 is $n(L_j)+2K\cdot(n(L_{j-1})+\sum_{z\in L_{j-1}} d'(z))$. So level L_j is emptied during the jth iteration of the "for" loop in $O(n(L_j)+n(L_{j-1})+\sum_{z\in L_j\cup L_{j-1}} d(z))$. Consequently, each iteration of the "while" loop checks one component H of G in $O(n(H)+m(H))$, from which the linearity of the whole algorithm follows.

We now return to our initial problem, the maximization of $\sum_{i=1}^{n} w_i x_i$ subject to $f(x)=0$, in the case of a graphic Boolean function. Because of the correspondence between f and its associated graph G, any vector x such that $f(x)=0$ is the characteristic vector of a stable set of G. So problem (1) is equivalent to the maximum weighted stable problem in G where every vertex x_i is assigned the weight w_i.

Algorithm MAXSTABLE will solve this problem in $O(n)$ time, for a completely separable graph G given with its pruning sequence $s_2,\dots,s_n$. It is based on the following observations:

• If a is a pendant vertex of G and b is its neighbor, and if $w_a \geq w_b$, let S be a maximum weighted stable set of $G-a$. Then $(S-\{b\})\cup\{a\}$ is a maximum weighted stable set of G. If $w_a<w_b$, let S be a maximum weighted stable set of $G-a$ in which vertex b has weight $w_b'=w_b-w_a$. Then if $b\in S$, S is a maximum weighted stable set of G (with respect to the initial weights) and, if $b\notin S$, $S\cup\{a\}$ is a maximum weighted stable set of G.

• If x and y are true twins, any maximum weighted stable set of G contains at most one of them, and if it does it must be the heavier.

• If x and y are false twins, any maximum weighted stable set of G contains either none or both of them, so it is equivalent to a maximum weighted stable set of $G-x$ in which y has weight $w_y'=w_x+w_y$.

• The pruning sequence of $G-n$ is $(s_2,\dots,s_{n-1})$.

Algorithm MAXSTABLE($G;S,\sigma$).

Input. A completely separable graph G on n vertices given by its pruning sequence $(s_2,\dots,s_n)$, with vertex weights $w_1,\dots,w_n$.

Output. A stable set S of maximal weight, and its weight σ.

begin
  if $n=1$ **then** $S:=\{1\}$; $\sigma:=w_1$; **return**
  if $s_n=nPj$ and $w_n\geq w_j$ **then**
    call MAXSTABLE($G-n;S,\sigma$);
    if $j\in S$ **then** $S:=(S-\{j\})\cup\{n\}$; $\sigma:=\sigma-w_j+w_n$

else $S := S \cup \{n\}$; $\sigma := \sigma + w_n$
if $s_n = nPj$ and $w_n < w_j$ **then**
$w_j := w_j - w_n$; call MAXSTABLE$(G - n; S, \sigma)$;
$\sigma := \sigma + w_n$;
if $j \notin S$ **then** $S := S \cup \{n\}$
if $s_n = nFj$ **then**
$w_j := w_j + w_n$; call MAXSTABLE$(G - n; S, \sigma)$;
if $j \in S$ **then** $S := S \cup \{n\}$
if $s_n = nTj$ and $w_n \geq w_j$ **then**
$w_j := w_n$; call MAXSTABLE$(G - n; S, \sigma)$;
if $j \in S$ **then** $S := (S - \{j\}) \cup \{n\}$
if $s_n = nTj$ and $w_n < w_j$ **then** call MAXSTABLE$(G - n; S, \sigma)$
end

In the case of all weights equal to 1, the algorithm produces a maximum stable set S, and σ is the stability number of the graph. Since the graph is perfect, σ is also the clique-partition number.

The usual dual problem, that of the maximum weighted clique, can also be solved in O(n) time for a completely separable graph given with a pruning sequence. Algorithm MAXCLIQUE is based on the following observations:

- If a is a pendant vertex of G with neighbor b, a maximum weighted clique of G is either $\{a, b\}$ or a maximum weighted clique of $G - a$.
- If x and y are true twins, any maximum weighted clique of G contains either none or both of them, so it is equivalent to a maximum weighted clique of $G - x$ in which vertex y has weight $w'_y = w_x + w_y$.
- If x and y are false twins of G, then any maximum weighted clique of G contains at most one of them, and if it does it must be the heavier.

The implementation of Algorithm MAXCLIQUE is similar to that of MAX-STABLE and straightforward. In the case of all weights equal to 1, this algorithm produces a maximum clique of G whose size is also the chromatic number of the graph, since G is perfect.

Acknowledgement

The authors wish to thank Vašek Chvátal who read a first draft of this paper. Reference [7] has been communicated to us by Chvátal. We acknowledge the contribution of Caterina De Simone to an early stage of this work, and the valuable advice of Brigitte Jaumard on data structures. We thank the referees for their suggestions.

References

[1] H.-J. Bandelt and H.M. Mulder, Distance-hereditary graphs, J. Combin. Theory Ser. B 41 (1986) 182–208.

[2] C. Berge, Färbung von Graphen, deren samtliche bzw. deren ungerade Kreise starr sind (Zusammenfassung), Wiss. Z. Martin-Luther-Univ. Halle-Wittenberg Math.-Natur. Reihe 10 (1961) 114–115.

[3] R.E. Bixby, A composition for perfect graphs, in: C. Berge and V. Chvátal, eds., Topics on Perfect Graphs, Annals of Discrete Mathematics 21 (North-Holland, Amsterdam, 1984) 221–224.

[4] M. Burlet and J.P. Uhry, Parity graphs, in: A. Bachem, M. Grötschel and B. Korte, eds., Bonn Workshop on Combinatorial Optimization, Annals of Discrete Mathematics 16 (North-Holland, Amsterdam, 1982) 1–26.

[5] V. Chvátal, Perfectly orderable graphs, in: C. Berge and V. Chvátal, eds., Topics on Perfect Graphs, Annals of Discrete Mathematics 21 (North-Holland, Amsterdam, 1984) 63–65.

[6] V. Chvátal and P.L. Hammer, Aggregations of inequalities, in: Integer Programming, Annals of Discrete Mathematics 1 (North-Holland, Amsterdam, 1977) 145–162.

[7] D.G. Corneil, Y. Perl and L.K. Stewart, A linear recognition algorithm for cographs, SIAM J. Comput. 14 (1985) 926–934.

[8] W.H. Cunningham and J. Edmonds, A combinatorial decomposition theory, Canad. J. Math. 22 (1980) 734–765.

[9] C.P. Gabor, W.L. Hsu and K.J. Supowit, Recognizing circle graphs in polynomial time, Manuscript.

[10] C.T. Hoàng, PhD Thesis, McGill University, Montreal (1985).

[11] C.T. Hoàng and N. Khouzam, A class of brittle graphs, Tech. Rept., Department of Computer Science, Rutgers University, New Brunswick, NJ (1986).

[12] E. Howorka, A. characterization of distance-hereditary graphs, Quart. J. Math. Oxford Ser. (2) 28 (1977) 417–420.

[13] E. Olaru, Über die Überdeckung von Graphen mit Cliquen, Wiss. Z. Tech. Hochsch. Ilmenau 15 (1969) 115–120.

[14] S. Seinsche, On a property of the class of n-colorable graphs, J. Combin. Theory Ser. B 16 (1974) 191–193.