Annals of Operations Research 116, 71–77, 2002

On Clique-Transversals and Clique-Independent Sets

GUILLERMO DURÁN * and MIN CHIH LIN * {willy, oscarlin}@dc.uba.ar
Departamento de Computación, Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires, Buenos Aires, Argentina

JAYME L. SZWARCFITER ** jayme@nce.ufrj.br
Instituto de Matemática, NCE and COPPE, Universidade Federal do Rio de Janeiro, Caixa Postal 2324, 20001-970 Rio de Janeiro, RJ, Brazil

Abstract. A clique-transversal of a graph G is a subset of vertices intersecting all the cliques of G. A clique-independent set is a subset of pairwise disjoint cliques of G. Denote by $\tau_C(G)$ and $\alpha_C(G)$ the cardinalities of the minimum clique-transversal and maximum clique-independent set of G, respectively. Say that G is clique-perfect when $\tau_C(H) = \alpha_C(H)$, for every induced subgraph H of G. In this paper, we prove that every graph not containing a 4-wheel nor a 3-fan as induced subgraphs and such that every odd cycle of length greater than 3 has a short chord is clique-perfect. The proof leads to polynomial time algorithms for finding the parameters $\tau_C(G)$ and $\alpha_C(G)$, for graphs belonging to this class. In addition, we prove that to decide whether or not a given subset of vertices of a graph is a clique-transversal is Co-NP-Complete. The complexity of this problem has been mentioned as unknown in the literature. Finally, we describe a family of highly clique-imperfect graphs, that is, a family of graphs G whose difference $\tau_C(G) - \alpha_C(G)$ is arbitrarily large.

Keywords: clique-independent sets, clique-perfect graphs, clique-transversals, highly clique-imperfect graphs, integer linear programming, linear programming

AMS subject classification: 05C69, 05C50, 90C05, 90C10

1. Introduction

Let G be a finite undirected graph, $V(G)$ and $E(G)$ the vertex and edge sets of G, respectively. Denote $|V(G)| = n$ and $|E(G)| = m$.

A *clique* of G is a complete subgraph maximal under inclusion. A set of vertices that meets all the cliques of G is a *clique-transversal* of G. A *clique-independent set* is a collection of pairwise vertex-disjoint cliques. The *clique-transversal number* and *clique-independence number* of G, denoted $\tau_C(G)$ and $\alpha_C(G)$, are the sizes of the minimum clique-transversal and maximum clique-independent set of G, respectively. Clearly, $\tau_C(G) \geqslant \alpha_C(G)$, for any graph G. As defined in [11], a graph G is clique-perfect if $\tau_C(H) = \alpha_C(H)$, for every induced subgraph H of G. Strongly chordal graphs [6] and comparability graphs [4] are examples of clique-perfect graph classes.

* Partially supported by UBACyT Grants X036 and X127 and PID Conicet Grant 644/98, Argentina.
** Partially supported by the Conselho Nacional de Desenvolvimento Científico e Tecnológico, CNPq, and Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro, FAPERJ, Brazil.

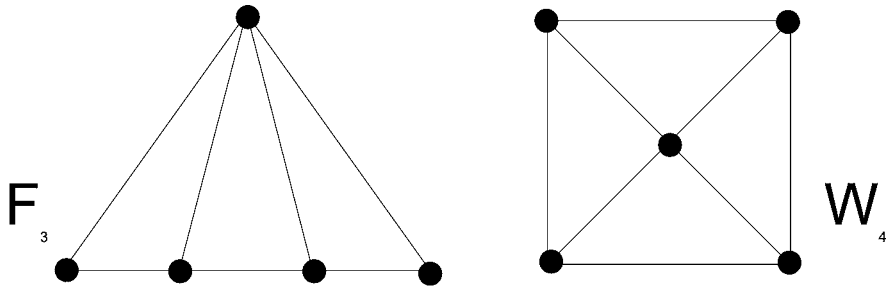

Figure 1. 3-fan and 4-wheel.

The *chromatic number* of a graph G is the smallest number of colours that can be assigned to the vertices of G in such a way that no two adjacent vertices receive the same colour. An obvious lower bound is the maximum cardinality of the cliques of G, the *clique number* of G, denoted by $\omega(G)$. Berge [5] proposed to call a graph G perfect whenever the chromatic number of every induced subgraph H of G is equal to $\omega(H)$. For more background information on perfect graphs see [10].

A *chord* of a cycle C of G is an edge joining two non consecutive vertices of C. If these vertices are at distance 2 in C then the chord is called *short*. Graphs with short chords have been previously considered in the context of perfect graphs [13,16].

Clearly, perfect graphs are not necessarily clique-perfect. On the other hand, clique-perfect graphs are not necessarily perfect, answering a question formulated in [11]. The graph $\overline{C_{6j+3}}$, the complement of a chordless cycle of length $6j + 3$, is clique-perfect but not perfect, for any $j \geqslant 1$ [15].

The problem of determining $\tau_C(G)$ is NP-hard [8], as well as that of finding $\alpha_C(G)$ [6]. Both problems can be solved in polynomial time for strongly chordal graphs [6], comparability graphs [4] and Helly circular-arc graphs [7,11].

Denote by F_3 a fan with 5 vertices and by W_4 a wheel with 5 vertices, respectively (see figure 1). Define $\mathcal{G}$ as the class of graphs which do not contain a F_3 nor a W_4 as induced subgraphs and such that every odd cycle of length $\geqslant 5$ has a short chord.

In the present paper, we address questions concerning clique-transversals and clique-independent sets. First, we prove that deciding whether a set of vertices of a graph G is a clique-transversal is Co-NP-Complete, a problem whose complexity is mentioned as open in [11]. Further, we show that the parameters $\tau_C(G)$ and $\alpha_C(G)$ can be computed in polynomial time for graphs belonging to the class $\mathcal{G}$. The method employs integer linear programming and also leads to the conclusion that $\mathcal{G}$ is a clique-perfect graph class. Finally, we describe a family of graphs such that the difference between τ_C and α_C is arbitrarily large.

2. The complexity of recognizing clique-transversals

In this section, we prove that the problem of deciding if a set of vertices of a graph is a clique-transversal is Co-NP-Complete.

Theorem 2.1. Given a graph G and a subset $S \subseteq V(G)$, the problem of deciding whether or not S is a clique-transversal of G is Co-NP-Complete.

Proof. Clearly, this decision problem is in Co-NP. A clique C of G which does not intersect a subset of vertices S is a certificate for S not to be a clique-transversal of G. Such a certificate can be recognized in polynomial time.

Let us see that the problem is NP-Hard. Transformation from the satisfiability problem. Let B be a boolean expression in conjunctive normal form, with clauses L_i, $1 \leqslant i \leqslant p$, each L_i having q_i literals. Construct a graph G, as follows. There is one vertex v_i of G, for each clause L_i. In addition, one vertex w_{ij}, for each occurrance of a literal in L_i, $1 \leqslant i \leqslant p$ and $1 \leqslant j \leqslant q_i$. The edges of G are the following. For all $1 \leqslant i, k \leqslant p$, $i \neq k$ and $1 \leqslant j \leqslant q_i$, $(v_i, v_k), (v_i, w_{kj}) \in E(G)$. Denote by ℓ_{ij} the literal of L_i, corresponding to w_{ij}. The edges (w_{ij}, w_{kt}) exist precisely when $i \neq k$ and $\ell_{ij} \neq \overline{\ell_{kt}}$. The construction of G is completed. Finally, the subset $S \subseteq V(G)$ of the above decision problem is defined as $S = \{v_1, \ldots, v_p\}$.

If B is satisfiable, let w_{ij_i} be the vertex of G corresponding to the literal of B, which satisfies clause L_i. In this case, $\{w_{1j_1}, \ldots, w_{pj_p}\}$ induces a clique of G which does not intersect S because $(v_i, w_{ij_i}) \notin E(G)$. Conversely, suppose that S does not intersect all cliques of G, and let C be a clique disjoint from S. Then C must contain one vertex w_{ij_i}, for each $1 \leqslant i \leqslant p$. This means that the set of literals $\{\ell_{1j_1}, \ldots, \ell_{pj_p}\}$ satisfies B. Consequently, B is satisfiable if and only if S is not a clique transversal of G. □

It follows from the above proof, that the problem remains Co-NP-Complete if S induces a clique of G.

3. Clique-transversals and clique-independent sets on the class $\mathcal{G}$

In this section, we prove that the clique-transversal number and the clique-independence number can be computed in polynomial time for graphs belonging to $\mathcal{G}$. In addition, we show that such graphs are necessarily clique-perfect.

Let G be a graph and $M_1, \ldots, M_k$ the cliques of G. Define A_G, the clique matrix of G, as a 0–1 matrix whose entry (i, j) is 1 if $v_j \in M_i$ and 0, otherwise. Let e_k be a vector with k 1's.

The problem of finding a minimum clique-transversal set can be formulated using integer linear programming (ILP) as follows:

$$\begin{aligned} &\text{Min } \sum_{i=1}^{n} x_i \\ &\text{s.a.} \quad A_G x \geqslant e_k, \\ &\qquad\quad x \in \{0, 1\}^n. \end{aligned}$$

On the other hand, in a similar way, the problem of determining a maximum clique-independent set can be also solved by integer programming, according to the following formulation.

$$\begin{aligned} &\text{Max} \sum_{i=1}^{k} y_i \\ &\text{s.a.} \quad A_G^t y \leqslant e_n, \\ &\qquad\quad y \in \{0, 1\}^k, \end{aligned}$$

where A_G^t is the transpose of A_G.

We know that, in general, if the extreme points of the polyhedra defined by the linear relaxation of a ILP problem are integers, then the original ILP problem can be solved applying linear programming to the linear relaxation of the original problem (and linear programming can be solved in polynomial time [12]).

We will prove that this is the case of the formulations of the minimum clique-transversal set problem and the maximum clique-independent set problem for graphs in the class $\mathcal{G}$. Moreover, we will show that the sizes of the corresponding linear programming problems are bounded by polynomials in the size of the the input graph $G \in \mathcal{G}$.

Previously, the following definition is needed. A 0–1 matrix M is *balanced* if it does not contain a vertex–edge incidence matrix of an odd cycle as a submatrix.

The following fundamental result was proved by Fulkerson, Hoffman and Oppenheim.

Theorem 3.1 [9]. If M is a balanced matrix, then the polyhedra defined as $P_1(M) = \{x \mid Mx \geqslant e_k, x \geqslant 0\}$ and $P_2(M) = \{x \mid Mx \leqslant e_k, x \geqslant 0\}$ have only integer extreme points.

In the sequel we show that the clique matrix of a graph belonging to $\mathcal{G}$ is necessarily balanced.

Theorem 3.2. Let G be a graph in $\mathcal{G}$. Then, A_G is a balanced matrix.

Proof. Suppose the theorem false. Then A_G contains a submatrix A' which is the incidence matrix of an odd cycle. Consider the following two cases.

1. A' is a 3×3 matrix. Denote by M_1, M_2, M_3 and v_1, v_2, v_3, respectively, the cliques and vertices of G corresponding to A'. Without loss of generality, A' is of the following form:

	v_1	v_2	v_3
M_1	1	1	0
M_2	0	1	1
M_3	1	0	1

Then the vertices v_1, v_2, v_3 induce a triangle in G. Since M_1 is a clique not containing v_3, there exists a vertex $w \in M_1$ adjacent to v_1 and v_2 and not to v_3. Similarly, there exists a vertex $z \in M_2$ adjacent to v_2 and v_3 and not to v_1. Consequently, the subgraph induced in G by $\{v_1, v_2, v_3, w, z\}$ is either isomorphic to W_4 or F_3, according whether or not w and z are adjacent, respectively. Both cases lead to contradictions.

2. A' is a $j \times j$ matrix, $j \geqslant 5$ and odd. Denote by $M_1, \ldots, M_j$ and $v_1, \ldots, v_j$ the cliques and vertices corresponding to A', respectively. Consequently, the vertices $v_1, \ldots, v_j$ form an odd cycle C on G. We may suppose that C is the cycle $v_1, \ldots, v_j, v_1$. Since $G \in \mathcal{G}$, we conclude that C has a short chord. Without loss of generality, suppose that the short chord joins v_1 and v_3. Therefore v_1, v_2 and v_3 induce a triangle in G, which is contained in some clique other than $M_1, \ldots, M_j$. As before, since M_1 and M_2 are cliques, there exists a vertex $w \in M_1$ which is adjacent both to v_1 and v_2 and not to v_3. Similarly, there exists a vertex $z \in M_2$ adjacent to v_2 and v_3 and not to v_1. The contradiction is the same as in the first case. □

Theorems 3.1 and 3.2 imply that the clique-transversal and clique-independence numbers of a graph $G \in \mathcal{G}$ can be determined by solving the linear relaxations of the described integer programming problems, respectively. In order to ensure a polynomial time bound it remains to show that the clique matrix A_G of G can be computed in polynomial time. However, this fact is a simple consequence of the following theorem.

Denote by pK_2 the graph formed by p disjoint copies of K_2.

Theorem 3.3 [14]. Let G be a graph which does not contain $\overline{pK_2}$ as induced subgraph for an integer p. So, the number of cliques of G is bounded by $n^{2(p-1)}$.

Corollary 3.1. Let G be a graph in $\mathcal{G}$. Then A_G can be computed in $O(n^5 m)$ time.

Proof. Clearly, $\overline{3K_2}$ contains a 4-wheel as an induced subgraph. So, the graph G can not contain $\overline{3K_2}$, by the definition of $\mathcal{G}$. Consequently, by theorem 3.3 G has at most n^4 cliques. The algorithm of [17] generates all cliques of a graph, using $O(nm)$ time per clique. Consequently, the collection of cliques of G can de obtained in $O(n^5 m)$ time. The corollary follows. □

Consequently,

Corollary 3.2. The problems of determining $\tau_C(G)$ and $\alpha_C(G)$ can be solved in polynomial time for graphs in $\mathcal{G}$.

Finally, the following simple argument shows that graphs belonging to $\mathcal{G}$ are necessarily clique-perfect.

Corollary 3.3. The class $\mathcal{G}$ is a clique-perfect graph class.

Proof. The clique matrix is balanced, so we can solve the problems of determining $\tau_C(G)$ and $\alpha_C(G)$ using the linear relaxations of the corresponding integer linear programming formulations. The linear relaxation of the second problem is the dual of the linear relaxation of the first one. It follows that the optimum value is the same for both relaxations. Finally, since that $\mathcal{G}$ is a hereditary class, $\tau_C(H) = \alpha_C(H)$ for any H induced subgraph of G, where G is a graph in $\mathcal{G}$. $\square$

4. Highly clique-imperfect graphs

A natural question is to look for a family of graphs, whose differences between their clique-transversal number and clique-independence number is arbitrarily large. In this section, we present a simple description of such a family of graphs.

Denote by F_t, $t \geqslant 1$, the graph obtained by the following construction. The vertices of F_t can be partitioned into a clique K_{2t+1} and an independent set I_j, having j vertices, where $j = \binom{2t+1}{t+1}$. Each vertex of I_j is adjacent precisely to a different subset of $t+1$ vertices of K_{2t+1}.

Theorem 4.1. For any $t \geqslant 1$, $\alpha_C(F_t) = 1$ and $\tau_C(F_t) = t + 1$.

Proof. There are $j + 1$ cliques in F_t. One of them is K_{2t+1} and each one of the other j cliques is formed by a vertex of I_j and a different subset of $t + 1$ vertices of K_{2t+1}. Clearly, it is not possible to choose two vertex-disjoint cliques, so $\alpha_C(F_t) = 1$. On the other hand, by selecting $t + 1$ vertices of K_{2t+1} all the cliques will be covered. Moreover, all of the $t + 1$ selected vertices are needed for the cover. Because, if we miss one of them there will be a clique formed by a vertex of I_j and a subset of $t + 1$ vertices of K_{2t+1} which is not covered. So, $\tau_C(F_t) = t + 1$. $\square$

The graph F_t satisfies $\tau_C(F_t) - \alpha_C(F_t) = t$, where t is an arbitrary integer. However, the size of F_t grows exponentially with t. It remains the question to describe a graph G_t, with a similar property with respect to the difference between these parameters, but whose size is polynomially bounded in t.

References

[1] T. Andreae, On the clique-transversal number of chordal graphs, Discrete Mathematics 191 (1998) 3–11.

[2] T. Andreae and C. Flotow, On covering all cliques of a chordal graph, Discrete Mathematics 149 (1996) 299–302.

[3] T. Andreae, M. Schughart and Z. Tuza, Clique-transversal sets of line graphs and complements of line graphs, Discrete Mathematics 88 (1991) 11–20.

[4] V. Balachandhran, P. Nagavamsi and C. Pandu Ragan, Clique transversal and clique independence on comparability graphs, Information Processing Letters 58 (1996) 181–184.

[5] C. Berge, Les problemes de colorations en théorie des graphes, Publ. Inst. Stat. Univ. Paris 9 (1960) 123–160.

[6] G. Chang, M. Farber and Z. Tuza, Algorithmic aspects of neighbourhood numbers, SIAM Journal on Discrete Mathematics 6 (1993) 24–29.
[7] G. Durán, Sobre grafos intersección de arcos y cuerdas en un círculo, Ph.D. Thesis, Universidad de Buenos Aires, Buenos Aires (2000) (in Spanish).
[8] P. Erdös, T. Gallai and Z. Tuza, Covering the cliques of a graph with vertices, Discrete Mathematics 108 (1992) 279–289.
[9] D. Fulkerson, A. Hoffman and R. Oppenheim, On balanced matrices, Mathematical Programming 1 (1974) 120–132.
[10] M. Golumbic, *Algorithm Graph Theory and Perfect Graphs* (Academic Press, New York, 1980).
[11] V. Guruswami and C. Pandu Rangan, Algorithmic aspects of clique-transversal and clique-independent sets, Discrete Applied Mathematics 100 (2000) 183–202.
[12] L. Khachian, A polynomial algorithm in linear programming, Soviet Math. Dokl. 20 (1979) 191–194.
[13] A. Lubiw, Short-chorded and perfect graphs, Journal of Combinatorial Theory Series B 51 (1991) 24–33.
[14] E. Prisner, Graphs with few cliques, in: *Graph Theory, Combinatorics and Applications: Proceedings of the 7th Quadrennial International Conference on the Theory and Applications*, eds. Y. Alavi and A. Schwenk (Wiley, New York, 1995) pp. 945–956.
[15] B. Reed, Personal communication (2000).
[16] L. Sun, Two classes of perfect graphs, Journal of Combinatorial Theory Series B 53 (1991) 273–291.
[17] S. Tsukiyama, M. Ide, H. Ariyoshi and Y. Shirakawa, A new algorithm for generating all the maximal independent sets, SIAM Journal on Computing 6 (1977) 505–517.
[18] Z. Tuza, Covering all cliques of a graph, Discrete Mathematics 86 (1990) 117–126.