On the hull number of some graph classes

Julio Araújo, Victor Campos, Frédéric Giroire, Leonardo Sampaio, Ronan Pardo Soares

On the hull number of some graph classes. [1]

J. Araujo [a,b] V. Campos [b] F. Giroire [a] L. Sampaio [a,2]
R. Soares [a,b]

[a] {*julio.araujo, frederic.giroire, leonardo.sampaio_rocha, ronan.pardo_soares*}*@inria.fr - MASCOTTE Project, I3S (CNRS & UNS) and INRIA, INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 Sophia-Antipolis Cedex France.*

[b] {*campos@lia.ufc.br*} *ParGO Research Group - Universidade Federal do Ceará - UFC - Campus do Pici, Bloco 910. 60455-760 - Fortaleza, CE - Brazil.*

Abstract

Given a graph $G = (V, E)$, the *closed interval* of a pair of vertices $u, v \in V$, denoted by $I[u, v]$, is the set of vertices that belongs to some shortest (u, v)-path. For a given $S \subseteq V$, let $I[S] = \bigcup_{u,v \in S} I[u, v]$. We say that $S \subseteq V$ is a *convex set* if $I[S] = S$.

The *convex hull* $I_h[S]$ of a subset $S \subseteq V$ is the smallest convex set that contains S. We say that S is a *hull set* if $I_h[S] = V$. The cardinality of a minimum hull set of G is the *hull number* of G, denoted by $hn(G)$.

We show that deciding if $hn(G) \leq k$ is an NP-complete problem, even if G is bipartite. We also prove that $hn(G)$ can be computed in polynomial time for cactus and P_4-sparse graphs.

Keywords: graph convexity, hull number, bipartite graph, cactus graph, P_4-sparse graph.

1 Introduction

In this paper, we adopt the graph terminology defined in [2]. All the graphs studied in this work are considered to be simple and undirected.

[1] Supported by the INRIA associated team EWIN.
[2] Partially supported by CAPES/Brazil and ANR Blanc AGAPE.

Given a graph $G = (V, E)$, the *closed interval* of a pair of vertices $u, v \in V$, denoted by $I[u, v]$, is the set of vertices that belong to some shortest (u, v)-path. If $S \subseteq V$, let $I[S] = \bigcup_{u,v \in S} I[u, v]$. We say that $S \subseteq V$ is a *convex set* if $I[S] = S$. Given a subset $S \subseteq V$, the *convex hull* $I_h[S]$ of S is the smallest convex set that contains S. We say that S is a *hull set* if $I_h[S] = V$. The cardinality of a minimum hull set of G is the *hull number* of G, denoted by $hn(G)$. The Hull Number problem is to determine whether $hn(G) \leq k$, for a given graph G and an integer k. This problem is NP-complete [5].

The Hull Number problem was defined in [7]. In [8], results of abstract convexity are shown to be valid for graph convexities. An oriented version of the Hull Number problem is studied in [4]. The hull number of a cartesian product of two connected graphs is characterized in [3]. Dourado et al. [6] show some bounds for the hull number of triangle-free graphs. In [5], it is also proved that the hull number of unit interval graphs, cographs and split graphs can be computed in polynomial time.

In Section 3, we answer an open question of Dourado et al. [5] by showing that it is NP-hard to compute the hull number of a given bipartite graph. We then present polynomial time algorithms to compute the hull number of cacti, in Section 4, and of P_4-sparse graphs in Section 5. This last result extends the previous known result for cographs [5]. Finally, we present some concluding remarks.

2 Preliminaries

In order to present our main results, it is necessary to first point out some observations on hull sets.

Lemma 2.1 (**[5]**) *Every simplicial vertex of a graph G belongs to any hull set of G.*

Lemma 2.2 (**[5]**) *If a graph G is not complete, then every universal vertex u of G does not belong to any minimum hull set of G.*

Lemma 2.3 (**[5]**) *Let G be a graph and H be an isometric subgraph of G. Then for every hull set S of H it holds that $V(H) \subseteq I_h(S)_G$.*

Lemma 2.4 (**[5]**) *Let G be a graph and S a proper and non-empty subset of $V(G)$. If $V(G) \backslash S$ is convex, then every hull set of G contains at least one vertex of S.*

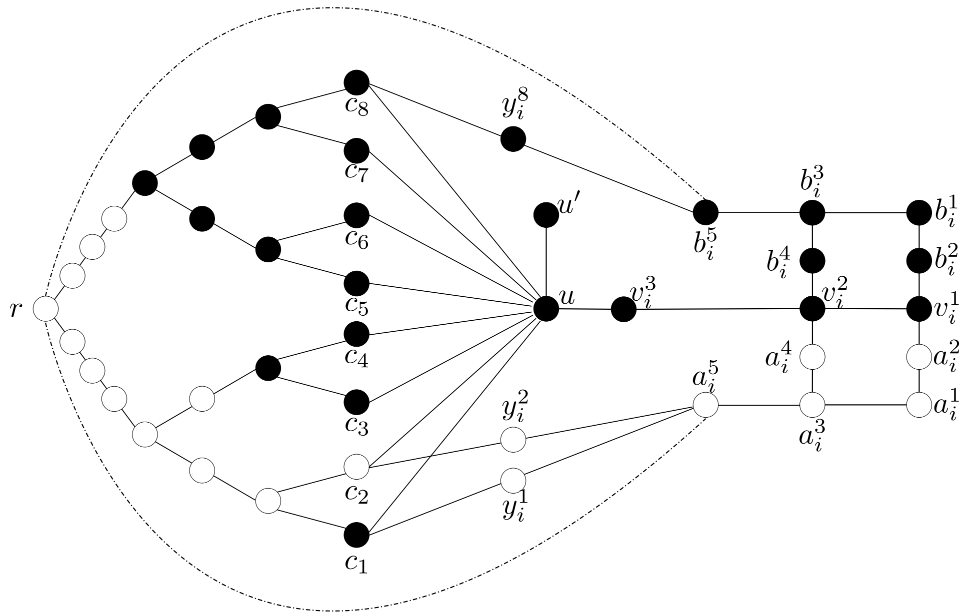

Fig. 1. Subgraph of the bipartite instance $G(\mathcal{F})$ containing the gadget of a variable x_i that appears positively in clauses C_1 and C_2, and negatively in C_8. If x_i appears positively in C_j, link a_i^6 to c_j through y_i^j. If it appears negatively, we use b_i^6 instead of a_i^6. See the appendix for more details.

3 Bipartite graphs

In this section, we study the complexity of the Hull Number problem for bipartite graphs.

Theorem 3.1 *Given a bipartite graph G and an integer k, deciding if $hn(G) \leq k$ is an NP-complete problem.*

Sketch of the proof: This proof is a modification of the one presented for general graphs [5]. We reduce SAT to Hull Number in bipartite graphs by creating an instance as depicted in Figure 1.

As it happens in the proof in [5]: a given SAT formula $\mathcal{F}$ is satisfied if, and only if, the hull number of the corresponding bipartite instance is at most $n+1$, where n is the number of variables of $\mathcal{F}$.

$\square$

Now, we show some *approximability results.* Let $IG(G)$ be the *incidence graph* of G, obtained from G by adding one vertex s_{uv}, for each edge $uv \in E(G)$, and replacing the edge uv by the edges $us_{uv}, s_{uv}v$.

Proposition 3.2 *$hn(IG(G)) \leq hn(G) \leq 2hn(IG(G))$.*

Corollary 3.3 *If there exists a k-approximation algorithm B to compute the hull number of bipartite graphs, then B is a $2k$-approximation algorithm for any graph.*

4 Cacti graphs

It is easy to observe that the hull number of a tree is equal to its number of leaves. In this section, we generalize this simple result to the class of cacti. A *cactus* is a graph in which every pair of cycles have at most one common vertex. In order to compute the hull number of a cactus, we show how to select the vertices of a minimum hull set.

Lemma 4.1 *Let C be an induced cycle in G. If C has at most one vertex v with degree strictly more than 2, then any minimum hull set of G contains either one or two vertices of $V(C)$.*

Lemma 4.2 *Let C be an induced cycle in G. Let $v_1, \ldots, v_k$ be the vertices of C having degree greater than two, for $k \geq 2$. If there is a set of vertices S of C such that they are not in any shortest (v_i, v_j)-path, $1 \leq i, j \leq k$, in C, then any minimum hull set of G contains exactly one vertex of S.*

Theorem 4.3 *Given a cactus G, $hn(G)$ can be found in linear time.*

Proof. By a simple breadth-first search all the cycles of a cactus can be detected. For each one, it is necessary to detect if it has at most one vertex of degree greater than two. In case it has at least two vertices of degree greater than two, it is necessary to also verify if there is a set S satisfying Lemma 4.2. Finally, every vertex of degree one must be chosen. These tests can be done in linear time and the result follows by Lemmas 2.1, 4.1 and 4.2.

□

5 P_4-sparse graphs

A graph is said to be a *cograph* if it is P_4-free. Computing the hull number of cographs can be done in polynomial time [5]. In this section, we extend this result to the class of P_4-sparse graphs which properly contains the cographs.

A graph is P_4-sparse if, for every $S \subseteq V$ such that $|S| = 5$, $G[S]$ contains at most one induced P_4 . Clearly, a cograph is a P_4-sparse graph. We call G a *strict* P_4-sparse graph if G is P_4-sparse and contains at least one P_4 as induced subgraph.

The algorithm we present to compute the hull number of any P_4-sparse graph G uses its modular decomposition tree $T(G)$. The idea is to perform a post-order traversal in $T(G)$ calculating $hn(G)$. The definition and classical results about the modular decomposition of graphs can be found in [10].

It is known that the neighborhood nodes of the modular decomposition of P_4-sparse graphs are spiders [9]. A *spider* is a graph whose vertex set can be partitioned into disjoint sets S, K, and R such that: $|S| = |K| \geq 2$, S is a stable set, K is a clique; every vertex in R is adjacent to all the vertices in K and to no vertex in S; there exists a bijection $f : S \longrightarrow K$ such that, for all vertices $s \in S$, either $N_G(s) \cap K = K - \{f(s)\}$ (*thick* spider) or $N_G(s) \cap K = \{f(s)\}$ (*thin* spider).

Note that R may be empty, it is the only possible non-trivial strong submodule of a spider, and that if $|S| = |K| = 2$, the spider is, by definition, thin and thick. However, in order to simplify the notation in the proofs, *we consider that a spider is called thick if, and only if,* $|S| = |K| \geq 3$. Moreover, we remark that it is possible to find a modular decomposition tree in such a way that the series and parallel nodes of the tree have only two children each.

Finally, we present how to compute the hull number of each inner node of the modular decomposition tree of a P_4-sparse graph. First, we need the following lemma:

Lemma 5.1 *Let $G = (V, E)$ be an arbitrary graph such that V can be partitioned into three sets: a clique K, a non-empty set R which is not a clique, and $S = V \backslash (K \cup R)$ with the property that, for every vertex $v \in R$, we have $(N(v) \cap (V \backslash R)) = K$. Then any minimum hull set of G contains a minimum hull set of the subgraph $H \subseteq G$, induced by $K \cup R$.*

Lemma 5.2 ([5]) *If $G = G_1 \cup G_2$, for any two graphs G_1 and G_2, then $hn(G) = hn(G_1) + hn(G_2)$.*

Lemma 5.3 *Let G_1 and G_2 be two P_4-sparse graphs and $G = G_1 \oplus G_2$. Then, $hn(G)$ can be computed in polynomial-time.*

Lemma 5.4 *Let $G = (S \cup K \cup R, E)$ be a spider. Then,*

$$hn(G) = \begin{cases} |S| + hn(G[K] \oplus G[R]), & \text{if } R \neq \emptyset \text{ and } R \text{ is not a clique;} \\ |S| + |R|, & \text{otherwise.} \end{cases}$$

Theorem 5.5 *Given a P_4-sparse graph G, the hull number of G can be found in polynomial time.*

Proof. The modular decomposition tree of any graph G can be found in linear time [11].

The complexity of the post-order traversal is $\mathcal{O}(n)$. For each inner node, the algorithm may spend $\mathcal{O}(n)$ operations in case 2 of Lemma 5.3 or when $R \neq \emptyset$ in Lemma 5.4. Consequently, the complexity of the algorithm is $\mathcal{O}(n^2)$. $\square$

6 Conclusions

In this paper, we simplified the reduction of Dourado et al. [5] to answer two questions they asked about the complexity of computing the hull number of triangle-free graphs and, in particular, of bipartite graphs. However, the complexity of this problem for interval graphs and chordal graphs remains open.

Finally, we used the modular decomposition tree of P_4-sparse graphs to compute the hull number of these graphs. The P_4-sparse graphs are contained in the class of $(q, q-4)$-graphs for which every set of q vertices induce at most $q-4$ disjoint P_4's [1]. It would be nice to extend our results to this class of graphs.

References

[1] Babel, L. and S. Olariu, *On the structure of graphs with few p4s*, Discrete Appl. Math. **84** (1998), pp. 1–13.

[2] Bondy, J. A. and U. S. R. Murty, "Graph Theory," Graduate Texts in Mathematics, Springer, 2008.

[3] Cagaanan, G. B., S. R. Canoy and Jr., *On the hull sets and hull number of the cartesian product of graphs*, Discrete Mathematics **287** (2004), pp. 141 – 144.

[4] Chartrand, G., J. F. Fink and P. Zhang, *The hull number of an oriented graph*, International Journal of Mathematics and Mathematical Sciences **2003** (2003), pp. 2265–2275.

[5] Dourado, M. C., J. G. Gimbel, J. Kratochvíl, F. Protti and J. L. Szwarcfiter, *On the computation of the hull number of a graph*, Discrete Mathematics **309** (2009), pp. 5668 – 5674, combinatorics 2006, A Meeting in Celebration of Pavol Hell's 60th Birthday (May 1-5, 2006).

[6] Dourado, M. C., F. Protti, D. Rautenbach and J. L. Szwarcfiter, *On the hull number of triangle-free graphs*, SIAM J. Discret. Math. **23** (2010), pp. 2163–2172.

[7] Everett, M. G. and S. B. Seidman, *The hull number of a graph*, Discrete Mathematics **57** (1985), pp. 217 – 223.

[8] Farber, M. and R. E. Jamison, *Convexity in graphs and hypergraphs*, SIAM J. Algebraic Discrete Methods **7** (1986), pp. 433–444.

[9] Giakoumakis, V. and J. Vanherpe, *On extended P_4-reducible and extended P_4-sparse graphs*, Theoretical Computer Science **180** (1997), pp. 269–286.

[10] McConnell, R. M. and J. P. Spinrad, *Modular decomposition and transitive orientation*, Discrete Mathematics **201** (1999), pp. 189–241.

[11] Tedder, M., D. G. Corneil, M. Habib and C. Paul, *Simpler Linear-Time Modular Decomposition Via Recursive Factorizing Permutations*, in: *Proceedings of the 35th International Colloquium on Automata, Languages and Programming (ICALP)*, Lecture Notes in Computer Science **5125**, 2008, pp. 634–645.