Finding a Maximum-Weight Convex Set in a Chordal Graph

Jean Cardinal[1] *Jean-Paul Doignon*[1] *Keno Merckx*[1]

[1]Université libre de Bruxelles, Belgium

Abstract

We consider a natural combinatorial optimization problem on chordal graphs, the class of graphs with no induced cycle of length four or more. A subset of vertices of a chordal graph is (monophonically) convex if it contains the vertices of all chordless paths between any two vertices of the set. The problem is to find a maximum-weight convex subset of a given vertex-weighted chordal graph. It generalizes previously studied special cases in trees and split graphs. It also happens to be closely related to the closure problem in partially ordered sets and directed graphs. We give the first polynomial-time algorithm for the problem.

E-mail addresses: jcardin@ulb.ac.be (Jean Cardinal) doignon@ulb.ac.be (Jean-Paul Doignon)
kmerckx@ulb.ac.be (Keno Merckx)

1   Introduction

In many practical optimization problems, feasible solutions consist of one or more sets that are required to satisfy some kind of convexity constraint. They can take the form of geometrically convex sets, such as in spatial planning problems [40], electoral district design [28], or underground mine design [35]. Alternatively, convexity can be defined in a combinatorial fashion.

In the closure problem [37], we are given a directed graph with (positive or negative) vertex weights, and we are asked to find a maximum-weight vertex subset with no outgoing edges. In the case where the directed graph is acyclic, this amounts to find a maximum-weight downset of a partial order. Here, convexity is interpreted as the property of being downward closed. Again, many practical applications are related to the closure problem. For instance, military targeting [33], transportation network design [38] and job scheduling [39]. Recently, a parametric version of the closure problem has been studied by Eppstein [13].

In their seminal paper, Farber and Jamison [16] developed the foundations of a combinatorial abstraction of convexity in graphs. In particular, they defined convex sets in graphs as subsets of vertices which contain the vertices of all chordless paths between any two vertices of the subset. This particular way of defining convexity in a graph is referred to as *monophonic convexity*. The collection of monophonic convex sets of a graph has specific nice properties and forms a "convex geometry" exactly if the graph is chordal. We consider the problem of finding a maximum-weight convex subset of a vertex-weighted chordal graph. We give a polynomial-time algorithm to solve the problem. Until now, only the special cases of trees [1, 29] and split graphs [5] were known to be polynomial-time solvable.

Our algorithm for chordal graphs makes use of an algorithm due to Picard [37] for the similar problem on ordered sets. Its design relies on a better understanding of the structure of a chordal graph from the point of view of its convex geometry. The results can be seen as a generalization of two algorithmic results for trees and split graphs, to all chordal graphs.

1.1   Previous works

The notion of a convex geometry appears in various contexts in mathematics and computer science. Dilworth [9] first examined structures very close to convex geometries in terms of lattice theory. The convex geometries were formally introduced by Jamison [25, 26] and Edelman and Jamison [11] in respectively 1981, 1982 and 1985. Later, Korte, Lovász and Schrader [30] considered antimatroids, which is the dual concept to the one of a convex geometry, as a subclass of greedoids. Today, the concept of a convex geometry (or antimatroid) appears in many fields of mathematics such as formal language theory (Boyd and Faigle [4]), choice theory (Koshevoy [31]), game theory (Algaba [2]) and mathematical psychology (Falmagne and Doignon [15]) among others.

When weights are assigned to the points of the convex geometry, the natural question of finding a convex set with maximum-weight arises. Particular subproblems are the closure problem [37], the maximum-weight subtree problem [1, 29], the maximum-weight path-closed set [21], or in a more geometrical setting, some variants of the minimum k-gons problem [14]. A more recent example is the problem of finding a maximum-weight convex set in a split graph [5]. For all of these problems, polynomial-time algorithms were published. We also mention that, without focusing on algorithms, Korte and Lovász [29] give linear characterizations of the convex set polytope for certain classes of antimatroids.

Searching for a general efficient algorithm to obtain a maximum-weight convex set in convex geometries seems hopeless because the problem is NP-hard even for special cases, see Eppstein [12] and Cardinal, Doignon and Merckx [5]. However, searching for a polynomial-time algorithm for certain classes of convex geometries could reveal bridges between mathematical areas and lead to better understanding of the underlying mathematical objects.

Chordal graphs and their representations have generated an extensive literature. See for instance Blair and Peyton [3], McKee and McMorris [32] or Golumbic [20] for theoretical and practical applications in various fields such as computational biology, phylogenetic, database, sparse matrix computation and statistics. But, despite a significant number of results about chordal graphs, there was to our knowledge, no polynomial-time algorithm to find a maximum-weight convex set.

1.2   Structure of the paper

In the next section, we give basic definitions and notation regarding convex geometries, graphs and posets, and formally define the optimization problem we consider. We also give the definition of the clique-separator graph of a chordal graph, which will be instrumental in what follows. In Section 3, we give a procedure solving the problem in a special family of instances. For this family, the problem is reduced to the closure problem in a partially ordered set. In Section 4 we generalize the algorithm to handle arbitrary chordal graphs and argue that it runs in polynomial time. In Section 5, we compute the time complexity of our algorithm and in Section 6 we give a detailed example of the execution on a small graph.

2   Preliminaries

We review here some basic notation and results for graphs and convex geometries, we also formally define the problems we investigate.

2.1   Notation for graphs

A (simple) graph G is a pair (V, E) where V is the (finite) set of vertices and E the set of edges, for a background on graph theory we recommend the book

by Diestel [8]. A *path* is a sequence of distinct vertices $(v_1, \ldots, v_n)$ such that $\{v_i, v_{i+1}\} \in E$ for all i in $\{1, \ldots, n-1\}$. The path is chordless (or *induced*) if no two vertices are connected by an edge that is not in the path. From a path $p = (v_1, \ldots, v_n)$ we can *extract* a chordless path by taking a shortest path between v_1 and v_n in the subgraph induced by the vertices in p. The graph is *connected* if for any u, v in V there is a path $(u, \ldots, v)$. A *connected component* of G is a maximal connected subgraph of G. Each vertex belongs to exactly one connected component, as does each edge. A *cycle* is a path $(v_1, \ldots, v_n)$ such that $\{v_n, v_1\}$ is an edge. A cycle is chordless (or *induced*) if no two vertices of the cycle are connected by an edge that does not itself belong to the cycle. A graph is *chordal* if every chordless cycle in the graph has at most three vertices. For $V' \subseteq V$ we denote by $N(V')$ the set of vertices w in $V \setminus V'$ such that $\{w, v\} \in E$ for some v in V'. We write $N(v)$ for $N(\{v\})$.

A *clique* K of G is a set of pairwise adjacent vertices, we say that K is a *maximal clique* if there is no clique K' of G such that $K \subset K'$. We denote by $\mathcal{K}_G$ the set of all maximal cliques in G. A *separator* S of G is a set of vertices such that there exist two vertices u, v in $V \setminus S$ connected by a path in the graph but not in $G - S$. We say that S is a *minimal separator* if there is no separator S' of G such that $S' \subset S$. For u, v in V, a subset S of $V \setminus \{u, v\}$ is a *uv-separator* if u and v are connected in G but not in $G - S$. The set S is a *minimal vertex separator* of G if S is a uv-separator for some u, v in V and S does not strictly contain any uv-separator. Note that any minimal separator is also a minimal vertex separator, but the converse does not hold in general. We denote by $\mathcal{S}_G$ the set of all minimal vertex separators in G. Note that in chordal graphs, every minimal vertex separator is a clique. We observe that for any chordal graph $G = (V, E)$ we have $|\mathcal{K}_G| \leqslant |V|$ and $|\mathcal{S}_G| \leqslant |V| - 1$, the proofs of those inequalities can be found in Fulkerson and Gross [18], and Ho and Lee [22] respectively.

2.2   Convex geometries on posets and chordal graphs

A set system $(V, \mathcal{C})$, where V is a finite set of elements and $\mathcal{C} \subseteq 2^V$, is a *convex geometry* when

$$\varnothing \in \mathcal{C},$$
$$\forall C_1, C_2 \in \mathcal{C} : C_1 \cap C_2 \in \mathcal{C},$$
$$\forall C \in \mathcal{C} \setminus \{V\}, \exists c \in V \setminus C : C \cup \{c\} \in \mathcal{C}.$$

The *convex sets* of the convex geometry $(V, \mathcal{C})$ are the members of $\mathcal{C}$. The *feasible sets* are the complements in V of the convex sets. An *antimatroid* (or *learning space* [15]) is a pair $(V, \mathcal{F})$ such that $(V, \mathcal{F}^\complement)$ is a convex geometry where $\mathcal{F}^\complement = \{V \setminus F : F \in \mathcal{F}\}$. All results on antimatroids have their counterpart for convex geometries.

We recall that a *partially ordered set* (or *poset*) P is a pair $(V, \leqslant)$ formed of a finite set V and a binary relation $\leqslant$ over V which is reflexive, antisymmetric, and transitive. For a poset $(V, \leqslant)$ an *ideal* I is a subset of V such that for all

elements a in I and b in V, if $b \leqslant a$, then b is also in I. The ideals are also known as *downsets*. We call idl(P) the set of ideals in P. For u, v in V, we say that v *covers* u in P with $u \neq v$, if $u \leqslant v$ and there is no x in $V \setminus \{u, v\}$ such that $u \leqslant x \leqslant v$.

One particular class of convex geometries described by Farber and Jamison [16] comes from the ideals of a poset. More precisely, let $(V, \leqslant)$ be a poset, then $(V, \mathrm{idl}(V, \leq))$ is a convex geometry called a *downset alignment*. Thus the convex sets in $(V, \mathrm{idl}(V, \leq))$ are the ideals in $(V, \leqslant)$. The downset alignments are the only convex geometries closed under union.

For a graph $G = (V, E)$, a set C of vertices is a *monophonically convex set* (*m-convex set*, or *convex set*) if C contains every vertex on every chordless path between vertices in C. We denote with $\mathcal{C}_G$ the set of m-convex sets of G. It happens that $(V, \mathcal{C}_G)$, is a convex geometry if and only if G is chordal, see Farber and Jamison [16]. The latter paper refers to another characterization of convex geometries than the one we mentioned; for a proof of the equivalence, see Edelman and Jamison [11]. For recent general results on the monophonic convexity, we refer the reader to Changat and Mathew [7], for the computation of some parameters of the convexity to Kannan and Changat [27] and for related algorithmic results to Dourado *et al.* [10]. For more details about m-convex sets and convex sets based on shortest paths, see Farber and Jamison [17].

Many classical problems in combinatorial optimization have the following form. For a set system $(V, \mathcal{C})$ and for a function $w : V \to \mathbb{R}$, find a set C of $\mathcal{C}$ maximizing the value of

$$w(C) = \sum_{c \in C} w(c).$$

For instance, the problem is known to be efficiently solvable for the system of independent sets of a matroid, thanks to the the greedy algorithm (see Oxley [34]). Since convex geometries capture a combinatorial abstraction of convexity in the same way as matroids capture linear dependence, the question of finding a convex set of maximum-weight arises naturally.

The problem of finding efficiently a maximum-weight convex set in a poset was solved by Picard [37]. The described algorithm calls as a subroutine a maximum flow algorithm (for instance Goldberg and Tarjan [19]) and runs in $O(mn \log(\frac{n^2}{m}))$ time, where n and m are respectively the number of elements and the number of cover relations in the poset.

2.3   The clique-separator graph for chordal graphs

Ibarra [24] introduces the *clique-separator graph* for chordal graphs. For a chordal graph G, he defines a mixed graph where the nodes are the maximal cliques and minimal vertex separators of G. Moreover, the (directed) arcs capture the inclusion-covering relation between the maximal cliques, while the (undirected) edges represent (minimal) inclusion from minimal vertex separators to maximal cliques of G. More precisely, the clique-separator graph $\mathcal{G}$ of

a chordal graph G has a set of *clique nodes*, one for each maximal clique of G and a set of *separator nodes* one for each minimal vertex separator of G. The clique-separator graph has also a set A of edges and arcs defined a follow. Each arc (S, S') is from a separator node S to a separator node S' such that $S \subset S'$ and there is no separator node S'' such that $S \subset S'' \subset S'$. Each edge $\{K, S\}$ is between a clique node K and a separator node S such that $S \subset K$ and there is no separator node S' such that $S \subset S' \subset K$. Later in this work, we will denote by $Ar_{\mathcal{G}}$ the set of arcs in a clique-separator graph $\mathcal{G}$. Figure 1 gives an example of a clique-separator graph of a chordal graph. Two of the mains results obtained by Ibarra are the following theorems.

Theorem 1 (Ibarra [24]) *For a chordal graph $G = (V, E)$, constructing the clique-separator graph of G can be done in $O(|V|^3)$ time.*
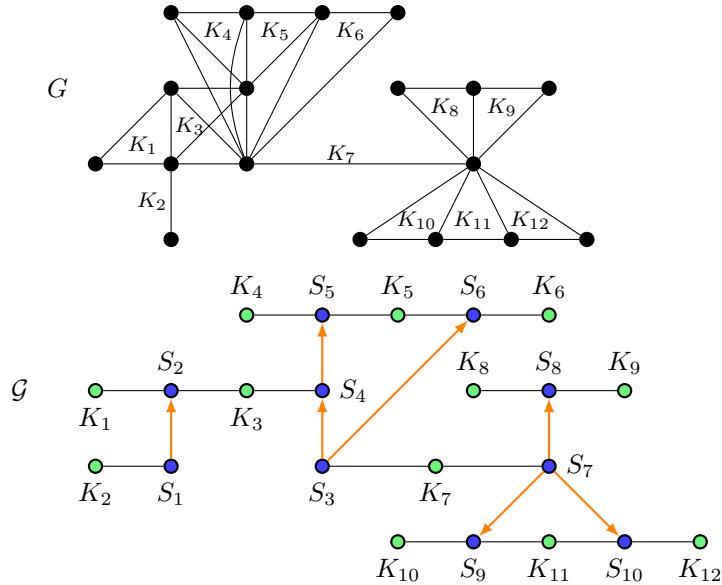

Figure 1: A clique-separator graph $\mathcal{G}$ of a chordal graph G.

Theorem 2 (Ibarra [24]) *Let $G = (V, E)$ be a chordal graph with clique-separator graph $\mathcal{G}$ and let S be a separator node of $\mathcal{G}$. If $G - S$ has connected components $G_1, \ldots, G_t$, then $t > 1$ and $\mathcal{G} - \{S' : S' \in \mathcal{S}_G, S' \subseteq S\}$ has connected components $\mathcal{G}_1, \ldots, \mathcal{G}_t$ such that for every $1 \leqslant i \leqslant t$, the vertex set of G_i is the same as the vertex set represented by the nodes of $\mathcal{G}_i - S$.*

2.4   Our main problem

Here is our main problem, the *maximum-weight convex set problem in chordal graphs*.

Problem 1 *Given a chordal graph G and a weight function $w : V \to \mathbb{R}$, find a set C in $\mathcal{C}_G$ that maximizes the value of $w(C)$.*

Our main result follows.

Theorem 3 *The maximum-weight convex set problem in chordal graphs can be solved in polynomial time.*

The well-known problem of finding a maximum-weight connected subtree in a tree can be solved by selecting a vertex as "root", finding a maximum-weight subtree that contains the root, and iterating this procedure for all possible roots (see Wolsey *et al.* [1]). In order to use a similar approach to solve Problem 1, we define a notion of root. It will be easier to work with chordal graphs which are connected. Note that our results straightforwardly extend to the non-connected case.

In order to simplify some of the later statements and arguments, we want to have in each maximal clique some vertex which is adjacent to no vertex outside the clique and which has weight zero. To this aim, we add such a vertex to any maximal clique (without changing the result of the optimization problems, see the end of the present subsection). Formally, let $G = (V, E)$ be a vertex-weighted graph. For each maximal clique K of G, we add a new vertex d_K to the graph and we make d_K adjacent to exactly the vertices in K (unless K already has vertices of weight zero, not adjacent to any vertex outside of K, then we select one of those that becomes d_k). The weight of d_K is set to 0, while the other vertices keep their weight. The resulting vertex-weighted graph is called the *extension G' of G*. Notice that the maximal cliques of G' are all of the form $K \cup \{d_K\}$, where K is a maximal clique of G; we call d_K the *dummy vertex* of the maximal clique $K \cup \{d_K\}$. Given a vertex-weighted chordal graph $G = (V, E)$, its extension $G' = (V', E')$ is also a vertex-weighted chordal graph. Remark that G and G' essentially have the same clique-separator graph. When $G = G'$, we say that the vertex-weighted chordal graph G is *extended*.

For a set R of vertices of an extended vertex-weighted chordal graph G, we say that a convex set C of $\mathcal{C}_G$ is *R-rooted* if $R \subseteq C$. If R is a singleton $\{r\}$ we write *r-rooted* instead of *$\{r\}$-rooted*. This modification allows us to define the following problem.

Problem 2 *Given an extended chordal graph G with a weight function $w : V \to \mathbb{R}$ and a maximal clique K of G, find a d_K-rooted convex set C of G that maximizes the value of $w(C)$.*

We show below that, given any vertex-weighted chordal graph G, solving Problem 2 for the extension G' of G for all K in $\mathcal{K}_G$ gives us a solution to Problem 1. The first lemma states the obvious link between the convex sets of G and G'.

Lemma 1 *Let $G = (V, E)$ be a chordal graph, C be a convex set of G and C' be a convex set of $G' = (V', E')$, the extension of G. Then C is a convex set of G' and $C' \cap V$ is a convex set of G.*

Proof: First, C is a convex set of G' because any chordless path in G' between two vertices of C is a chordless path in G. Second, $C' \cap V$ is convex in G because any chordless path in G between two vertices of $C' \cap V$ is a chordless path in G'. $\qquad\square$

The next lemma shows a stronger result than what we need for proving the equivalence between Problem 1 and Problem 2, but it will be useful.

Lemma 2 *Let $G = (V, E)$ be a chordal graph with a convex set C in G, and G' be the extension of G. Let K_C be a maximal clique of the graph induced by C. Then, for every K' in $\mathcal{K}_{G'}$ such that $K_C \subseteq K'$, the set $\{d_{K'}\} \cup C$ is convex in G'.*

Proof: For K' in $\mathcal{K}_{G'}$ such that $K_C \subseteq K'$, suppose that $\{d_{K'}\} \cup C$ is not convex in G'. So there is a chordless path $(d_{K'}, f_1, \ldots, f_t, c)$ in G' with c in C but $f_1, \ldots, f_t$ not in C. Because f_1 must be in K', we know that for all v in K_C we must have $\{v, c\} \in E$ (otherwise any chordless path in G we can extract from $(v, f_1, \ldots, f_t, c)$ contradicts the convexity of C). There results a contradiction with the maximality of K_C. $\qquad\square$

Lemmas 1 and 2 combined show that any algorithm solving Problem 2 in polynomial time establishes Theorem 3. Indeed, we run the algorithm solving Problem 2 on every maximal clique and save a maximum-weight solution C^* among all the outputs of the executions. Then we remove the dummy vertices from C^* and we are done.

In what follows, the chordal graphs we consider are extended: we consider that every maximal clique K contains a fixed, dummy vertex d_K.

3   A special case solvable in polynomial time

In this section, we solve Problem 2 for a family of special instances. We first define a partial order relation on the vertices of a given chordal graph. Then we use this relation to reduce instances of Problem 2 in this family to the closure problem in posets. The latter problem can be solved in polynomial time using Picard's algorithm [37].

3.1   The rooted poset

Let K be a maximal clique of a chordal graph $G = (V, E)$. We define the binary relation $\leqslant_K$ on V as the set of pairs $(u, v) \in V \times V$ such that there is a chordless path $(v, \ldots, d_K)$ that contains u. For the reduction we need to check that the relation is indeed a partial order.

Theorem 4 *For $G = (V, E)$ a chordal graph and K a maximal clique of G, the pair $(V, \leqslant_K)$ is a poset.*

It can be shown that the order relation we just defined is a special case of the C-factor relation defined by Edelman and Jamison [11] (taking the convex set C equal to $\{d_K\}$). The poset $P_K = (V, \leqslant_K)$ will be referred to as the K-rooted poset of G. Figure 3 shows a chordal graph and the Hasse diagram for $(V, \leqslant_K)$ with $K = \{1, 2, d_{\{1,2\}}\}$.

Theorem 4 directly follows from the two lemmas below, respectively stating that the relation is antisymmetric and transitive, and whose proofs are illustrated in Figure 2. The reflexivity of the relation is obvious.

Lemma 3 (Antisymmetry) *For $G = (V, E)$ a chordal graph and K a maximal clique of G, the relation $\leqslant_K$ is antisymmetric.*

Proof: For a and b in V, we show that we cannot have $a \leqslant_K b$, $b \leqslant_K a$ and $a \neq b$. Suppose $a \leqslant_K b$ and $b \leqslant_K a$, so there are two chordless paths $(u_1, \ldots, u_j, \ldots, u_n)$ and $(v_1, \ldots, v_l, \ldots, v_m)$ with $u_1 = v_l = a$, $u_j = v_1 = b$ and $u_n = v_m = d_K$. If we take the path

$$(u_{j+1}, \ldots, u_n, v_{m-1}, \ldots, v_l, u_2, \ldots, u_{j-1}),$$

we can extract a chordless path p with starting vertex u_{j+1} and ending vertex u_{j-1}. The path p has at least three vertices because $\{u_{j+1}, u_{j-1}\}$ is not in E. The vertex b is not in p because $a \neq b$, so we can add it to p. But then, a contradiction arises, because we obtain a chordless cycle with more than three vertices due to the fact that b only forms an edge with u_{j-1} and u_{j+1} among the considered vertices. $\square$

Lemma 4 (Transitivity) *For $G = (V, E)$ a chordal graph and K a maximal clique of G, the relation $\leqslant_K$ is transitive.*

Proof: For a, b and c three different vertices in V, suppose we have $a \leqslant_K b$ and $b \leqslant_K c$. So we have $(u_1, \ldots, u_j, \ldots, u_n)$ and $(v_1, \ldots, v_l, \ldots, v_m)$, two chordless paths with $u_1 = a$, $u_j = v_1 = b$, $v_l = c$ and $u_n = v_m = d_K$. From the path $(u_1, \ldots, u_j, v_2, \ldots, v_m)$ we extract a chordless path $(u_1, \ldots, u_x, v_y, \ldots, v_m)$. If $y \in \{1, \ldots, l\}$ then c is in the chordless path and we have $a \leqslant_K c$. If $y \in \{l + 1, \ldots, m\}$, then from the following path:

$$(u_{j+1}, \ldots, u_n, v_{m-1}, \ldots, v_y, u_x, \ldots, u_{j-1}),$$

we can extract a chordless path p that avoids b, with starting vertex u_{j+1} and ending vertex u_{j-1}. A similar argument to the proof of the previous lemma shows a contradiction. $\square$

3.2 A reduction to the maximum-weight ideal in poset problem

We now give a sufficient condition on a pair (G, K), where G is a chordal graph and K a maximal clique of G, for the existence of a one-to-one correspondence
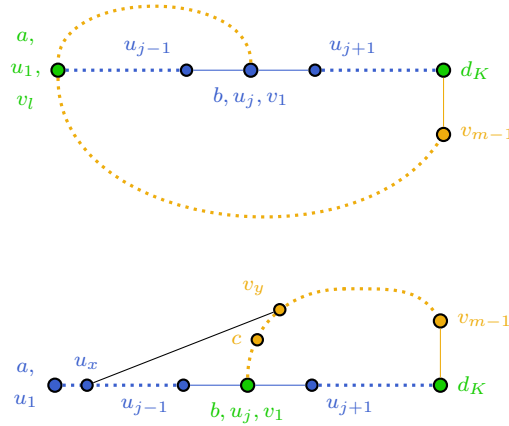
Figure 2: Illustrations of the proofs of antisymmetry and transitivity for the relation $\leqslant_K$.

between the nonempty ideals of the K-rooted poset and the d_K-rooted convex sets. Given a chordal graph G with clique-separator graph $\mathcal{G}$, for K in $\mathcal{K}_G$ and $a = (S_1, S_2)$ in $Ar_{\mathcal{G}}$, we say that a is K-*blocking* if S_1 is a minimal s_2d_K-separator for every s_2 in $S_2 \setminus S_1$. There is also an interpretation of the K-blocking property in the clique-separator graph. An arc (S_1, S_2) is K-blocking if $\mathcal{G} - \{S' : S' \in \mathcal{S}_G, S' \subset S_1\}$ has connected components $\mathcal{G}_1, \ldots, \mathcal{G}_t$ such that S_2 is included in $\mathcal{G}_i$ and K in $\mathcal{G}_j$ for some distinct i and j, and that there is no S in $\mathcal{G}_j$ such that (S, S_1) is an arc in $\mathcal{G}$. Figure 4 shows a clique-separator graph in which (S_1, S_2) and (S_6, S_7) are K_1-blocking arcs but (S_2, S_3) and (S_5, S_4) are not.

As shown in Theorem 5, the absence of K-blocking arcs is a sufficient condition for the correspondence between ideals of P_K and d_K-rooted convex sets. Figure 5 below gives a schematic view of the second part of the proof.

Theorem 5 *Let $G = (V, E)$ be a chordal graph with clique-separator graph $\mathcal{G}$, a maximal clique K in $\mathcal{K}_G$ such that there is no K-blocking arc in $\mathcal{G}$ and P_K be the K-rooted poset of G. Then a subset I of V is a nonempty ideal of P_K if and only if I is a d_K-rooted convex set in G.*

Proof: First, let C be a convex set containing d_K. For c in $C \setminus \{d_K\}$, any vertex u such that $u \leqslant_K c$ belongs to some chordless path. By convexity, we have $u \in C$ so C is an ideal of P_K.

Now let I be an ideal of P_K and suppose, for contradiction, that I is not convex. Then by definition, there must exist x and y in I and a chordless path $(x, f_1, f_2, \ldots, f_t, y)$ such that $f_1, f_2, \ldots, f_t$ do not belong to I. Note that x and y must be incomparable in P_K, as for otherwise f_1 or f_t would be contained in I. In particular, they are both different from d_K. Moreover we cannot have

$$\left(V, \leqslant_{\{1,2,d_{\{1,2\}}\}}\right)$$

$G = (V, E)$
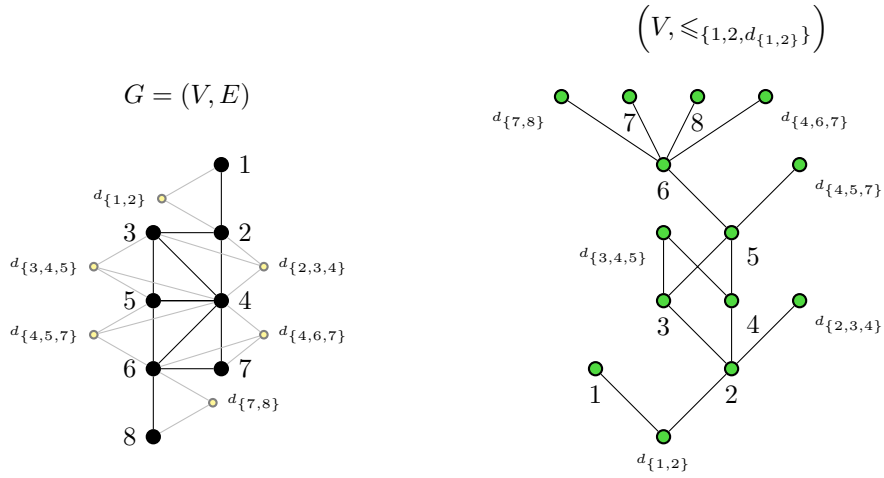
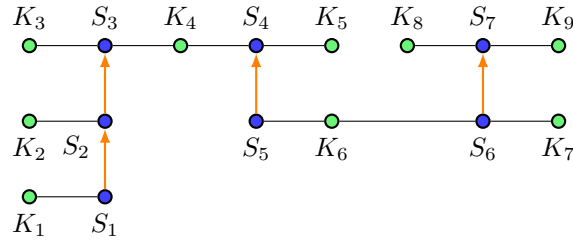Figure 3: A chordal graph and its $\{1, 2, d_{\{1,2\}}\}$-rooted poset.

Figure 4: A clique-separator graph with exactly two K_1-blocking arcs.

both $\{x, d_K\}$ and $\{y, d_K\}$ as edges, since otherwise $\{x, y\} \subseteq K$. So without loss of generality, we assume that $\{x, d_K\} \notin E$.

Let T be a minimal xy-separator included in the neighborhood $N(x)$ of x. Let $S = T \cap I$. We claim that S is either an xd_K-separator or a yd_K-separator. Suppose otherwise. Then there must be two chordless paths of the form $(x, u_1, u_2, \dots, u_n, d_K)$ and $(y, v_1, v_2, \dots, v_{n'}, d_K)$ contained in I and avoiding S. By concatenating them, we obtain a path from x to y in I avoiding S, which contradicts the fact that T was an xy-separator. This proves the claim.

In fact, S is an xd_K-separator because otherwise, we can extract a chordless path from $(y, f_t, \dots, f_1, x, \dots, d_K)$ that avoids S and contradicts the fact that S is an yd_K-separator. Now consider a minimal xd_K-separator $S_1 \subseteq S$ and a minimal vertex separator $S_2 \subseteq T$ such that $a = (S_1, S_2)$ is an arc of $\mathcal{G}$. We know such an arc exists because T is an xy-separator while S is not. We now show that a is K-blocking, a contradiction.

By definition, a is K-blocking if and only if S_1 is a td_K-separator for any $t \in S_2 \setminus S_1$. Suppose for contradiction that for some such t there exists a

chordless path from t to d_K avoiding S_1. We recall that $t \in T$ and $T \subseteq N(x)$, hence $\{x, t\}$ is in E. But then there is a chordless path from x to d_K avoiding S_1, contradicting that S_1 is an xd_K-separator. Hence a is indeed K-blocking. □
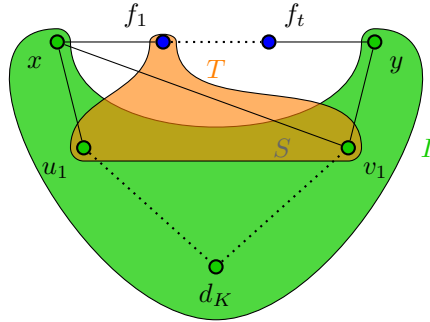

Figure 5: Illustration of the second part of proof for Theorem 5.

So whenever G has no K-blocking arc, it is possible to compute a maximum-weight d_K-rooted convex set of G in polynomial time by first computing the cover relation of the K-rooted poset, then using Picard's algorithm [37]. Note that the relation $\leqslant_K$ can be computed in polynomial time as we show later. There are some well-known examples of chordal graphs G such that for every K in $\mathcal{K}_G$, the clique-separator graph of G has no K-blocking arc. For example, k-trees have no arc in their clique-separator graph (see Patil [36] for details). We recall that a k-tree is a graph formed by starting with a clique of size $k+1$ and then repeatedly adding vertices with exactly k neighbors inducing a clique. In the next section, we will see how to deal with the case where the clique-separator graph contains a K-blocking arc.

4   A polynomial-time algorithm

We now consider chordal graphs G with one or more K-blocking arcs in their clique-separator graph, for some K in $\mathcal{K}_G$. We describe an algorithm for finding a maximum-weight convex set rooted in K.

For a chordal graph G with clique-separator graph $\mathcal{G}$ we define the subgraph $G \ominus a$ for $a = (S_1, S_2)$ in $Ar_{\mathcal{G}}$ as the graph induced by the union of S_1 and the connected component of $G - S_1$ that intersects S_2. Figure 6 shows an example of the $\ominus$ operation. Note that $G \ominus a$ is also a chordal graph (as any induced subgraph of a chordal graph is also chordal).

For a chordal graph $G = (V, E)$, a subset R of V and a weight function w, we denote by $\mathrm{opt}(G, R)$ a maximum-weight R-rooted convex set of G with respect to w. If R is a singleton $\{r\}$, we will write $\mathrm{opt}(G, r)$ instead of $\mathrm{opt}(G, \{r\})$. The algorithm proceeds in two main steps. In a first preprocessing phase, for each
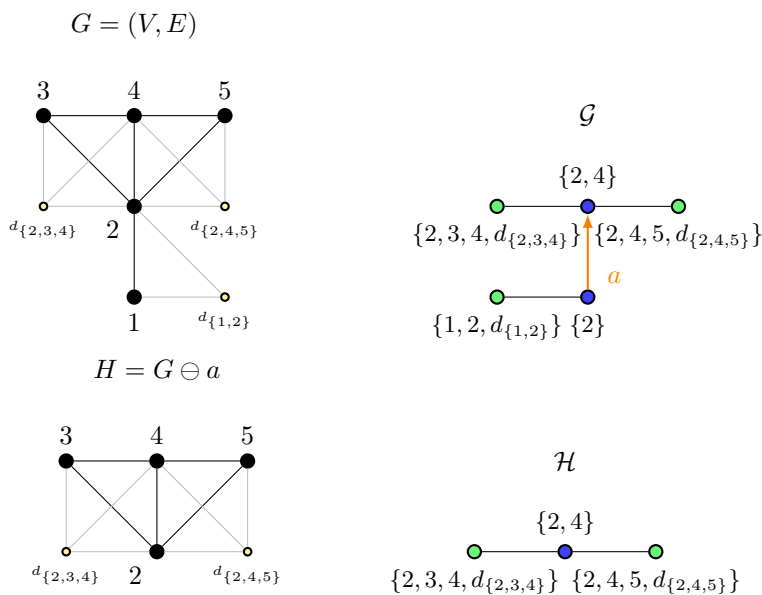
$$G = (V, E)$$

Figure 6: An example of the $\ominus$ operation.

arc $a = (S_1, S_2)$, we compute $\mathrm{opt}(G \ominus a, S_1)$ that is, a maximum-weight convex set of $G \ominus a$ rooted in the vertex separator S_1. After this preprocessing phase we denote by $\mathrm{label}(a)$ the solution of this subproblem. An algorithm for this preprocessing phase is described in Section 4.2.

4.1   Computation phase

In this second phase, we are going to use the labels of the arcs to compute a maximum-weight d_K-rooted convex. The algorithm proceeds essentially by collapsing the vertices of the subgraph $(G \ominus a) - S_1$ into a single vertex z_a for each arc $a = (S_1, S_2)$ that is K-blocking. The weight of z_a is then set to $w(\mathrm{label}(a)) - w(S_1)$, so that the weight of an optimal solution remains unchanged. This is detailed in Algorithm 1.

Note that the number of K-blocking arcs decreases at each iteration of the loop. Indeed, at least the vertex separator S_2 disappears. One step of the algorithm is illustrated by Figure 7. Since the goal is to find a maximum-weight convex set in the graph, we need to remember that including the vertex z_a in a solution for the collapsed instance amounts to choosing the set $\mathrm{label}(a) \setminus S_1$ in a solution of the original instance.

Theorem 6 *Let $G = (V, E)$ be a chordal graph with a maximal clique K and let $a = (S_1, S_2)$ be a K-blocking arc of $Ar_{\mathcal{G}}$. Let G^* be the graph obtained from G after applying Steps 2–4 of Algorithm 1 on a. Then $w(\mathrm{opt}(G, d_K)) = w(\mathrm{opt}(G^*, d_K))$.*

Algorithm 1: Finding a maximum d_K-rooted convex set in a chordal graph

---

Input: a chordal graph G and its clique-separator graph $\mathcal{G}$, a maximal clique K of G, a weight function w, the function label

Output: a maximum-weight K-rooted convex set C

1 while $\exists a = (S_1, S_2) \in Ar_{\mathcal{G}}$ *such that a is K-blocking* **do**

2 Identify the vertices of $(G \ominus a) - S_1$ into a new vertex z_a

3 $w(z_a) \leftarrow w(\text{label}(a)) - w(S_1)$

4 Add a dummy vertex to the new maximal clique $\{z_a\} \cup S_1$

5 Update $\mathcal{G}$

6 Use Picard's algorithm to compute a maximal weight d_K-rooted convex set C of G
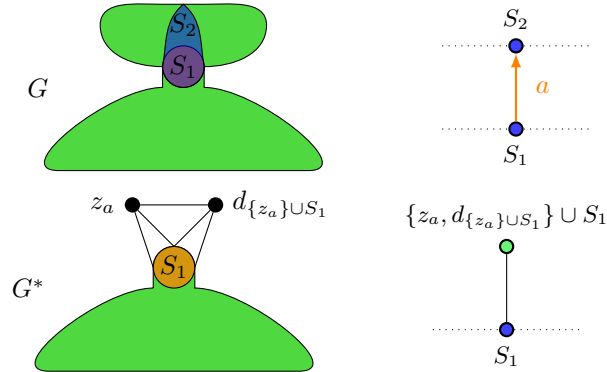
7 Return C

Figure 7: Illustration of the transformation in Algorithm 1 and the implication for the clique-separator graph.

Before proving Theorem 6, we make two simple observations.

Lemma 5 *Let G be a chordal graph, S be a minimal vertex separator of G and let V_1 and V_2 be the vertex sets of two distinct components of $G - S$. If C_1 and C_2 are two S-rooted convex sets in the graphs induced by $V_1 \cup S$ and $V_2 \cup S$ respectively, then $C_1 \cup C_2$ is a convex set of G.*

Proof: By contradiction, suppose there are c and c' in $C_1 \cup C_2$ and a chordless path $(c, f_1, \ldots, f_n, c')$ of G with $f_1, \ldots, f_n$ outside of $C_1 \cup C_2$. There must exist i in $\{1, \ldots, n\}$ such that f_i is in S otherwise we have a contradiction with the fact that S is a separator. But then, $f_i \in C_1 \cup C_2$ because $S \subseteq C_1 \cap C_2$, and we have a contradiction. □

Lemma 6 *Let $G = (V, E)$ be a chordal graph with a maximal clique K and let*

$a = (S_1, S_2)$ be a K-blocking arc of $Ar_{\mathcal{G}}$. Then, for a d_K-rooted convex set C in G that contains some vertex of $(G \ominus a) - S_1$, we have $S_1 \subset C$.

Proof: By contradiction, let s_1 be in $S_1 \setminus C$ and let c be in $C \cap (G \ominus a - S_1)$. We know that d_K and c are not in the same connected component of $G - S_1$. Because a is K-blocking, there is a chordless path $(d_K, v_1, \ldots, v_n, s_1, s_2)$ with $s_2 \in S_2 \setminus S_1$, $s_1 \in S_1$ and $v_n \notin S_1$. There is also a path $(s_1, s_2', \ldots, c)$ in $G \ominus a$ with $s_2' \in S_2 \setminus S_1$ from which we can extract a chordless path that only intersects S_1 in s_1. So we build a path $(d_K, v_1, \ldots, v_n, s_1, \ldots, c)$ that can not have a chord, a contradiction. $\square$

Proof: [Proof of Theorem 6] We decompose the equality into two inequalities. First, we show $w(\text{opt}(G, d_K)) \geqslant w(\text{opt}(G^*, d_K))$. More precisely, we show that for every d_K-rooted convex C^* of G^*, we have a d_K-rooted convex set C of G with $w(C) = w(C^*)$. If $z_a \notin C^*$ we take $C = C^*$ and we are done. Now, if $z_a \in C^*$, we define C as the union of $C^* \setminus \{z_a\}$ (which is convex because $N(z_a)$ induces a clique) with label(a). Obviously $w(C) = w(C^*)$, we only need to check that C is convex. Because a is K-blocking, $N(z_a) \setminus \{d_{\{z_a\} \cup S_1}\} = S_1$ is a minimal $d_K z_a$-separator, hence $C \setminus \{z_a\}$ must include S_1. We also know that S_1 is contained in label(a). From Lemma 5, C is convex in G, therefore $w(\text{opt}(G, d_K)) \geqslant w(\text{opt}(G^*, d_K))$.

We now show $w(\text{opt}(G, d_K)) \leqslant w(\text{opt}(G^*, d_K))$. More precisely, for every d_K-rooted convex C of G, we have a d_K-rooted convex set C^* of G^* with $w(C) \leqslant w(C^*)$. If C does not intersect $(G \ominus a) - S_1$, we take $C^* = C$ and we are done. If C does intersect $(G \ominus a) - S_1$, we define C^* as the union of z_a with the vertices of C that also are in G^*. We have $w(C) \leqslant w(C^*)$, otherwise we contradict the maximality of label(a). From Lemma 6, the vertices of C that are also in G^* form a convex set containing S_1. Since $\{z_a\} \cup S_1$ is a clique, hence is convex, Lemma 5 implies that C^* is also convex. So we have $w(\text{opt}(G, d_K)) \leqslant w(\text{opt}(G^*, d_K))$. $\square$

4.2   Preprocessing

We now describe the algorithm for computing the labels for the arcs in $\mathcal{G}$. This step is done only once and does not depend on the root of the convex set we are looking for. Recall that the label of an arc $a = (S_1, S_2)$ is the maximum-weight convex set of $G \ominus a$ rooted in S_1. Note that this algorithm uses Algorithm 1 as a subroutine on smaller graphs.

The algorithm is composed of two main ingredients. First, we need to label the arcs in an order such that the computation only involves arcs that are already labeled. We prove that we can achieve this by following the order of inclusion of the graphs $G \ominus a$. Second, in order to compute the optimal convex set rooted in S_1, we need to check all possible roots d_K such that S_1 is contained in K. This is detailed in Algorithm 2.

In step 6 of Algorithm 2, we can force S_1 to be in the solution C^* by assigning a sufficiently large weight to each vertex of S_1 before calling Algorithm 1. More

---

Algorithm 2: Labeling the arcs in $\mathcal{G}$

Input: a chordal graph G and its clique-separator graph $\mathcal{G}$, a maximal
        clique K of G, a weight function w
Output: the label function

1 while $\exists$ *an arc in* $Ar_{\mathcal{G}}$ *without label* **do**

2   Select $a = (S_1, S_2) \in Ar_{\mathcal{G}}$ without label such that every arc a' with
       $G \ominus a' \subset G \ominus a$ is already labeled

3 $M \leftarrow S_1$

4   Let $\mathcal{K}_a$ be the set of maximal cliques of G that contain S_1 and are
       contained in $G \ominus a$

5   **for** K *in* $\mathcal{K}_a$ **do**

6   Using Algorithm 1, compute a maximum-weight convex set C^*
         of $G \ominus a$ rooted in d_K and containing S_1

7 $M \leftarrow \max_w\{M, C^*\}$

8 $label(a) \leftarrow M$

---

precisely, we assign them the weight $\sum_{v \in V} |w(v)|$. By Lemma 2, looking for all the d_K-rooted convex sets with K in $\mathcal{K}_a$ ensures that we will find a maximum-weight S_1-rooted convex set of $G \ominus a$.

Note that Algorithm 2 labels the arcs in an order compatible with the partial order of inclusion of the graphs $G \ominus a$. The following lemma guarantees that the K-blocking arcs that will be processed by Algorithm 1 are all already labeled.

Lemma 7 *Let G be a chordal graph with clique-separator graph $\mathcal{G}$ and let $a = (S_1, S_2)$ in $Ar_{\mathcal{G}}$ and $a' = (S_3, S_4)$ an arc of the clique-separator graph of $G \ominus a$. If $G \ominus a' \not\subset G \ominus a$, then a' is not K-blocking for any maximal clique K in $\mathcal{K}_a$.*

Proof: Suppose that $G \ominus a' \not\subset G \ominus a$. We show that S_3 is not a minimal $s_4 d_K$-vertex separator for s_4 in $S_4 \setminus S_3$. If $G \ominus a' = G \ominus a$, then d_K is connected to s_4 in $(G \ominus a') - S_3$ and we have the result.

If $G \ominus a' \neq G \ominus a$, there is v in $G \ominus a'$ such that v is not in $G \ominus a$. So there must exist a chordless path p from s_4 to v that avoids S_3. But, because S_4 is in $G \ominus a$ and v is not, the path p must contains a vertex $s_1 \in S_1$. Now, because $\{s_1, d_K\}$ is an edge in G, we can deduce the existence of a path from s_4 to d_K that avoids S_3. $\qquad\square$

A complete execution of the algorithm on an example is given in Section 6.

5   Time complexity

From Algorithms 1 and 2, it seems straightforward that the time complexity needed to solve the maximum-weight convex set problem on a chordal graph $G = (V, E)$ is bounded by a polynomial in $|V|$ and $|E|$. More precisely, we see that the complexity of the algorithm used to solve Problem 1 on G will

be bounded by that of the preprocessing step. Indeed, the preprocessing step involves $|V||E|$ calls to Picard's algorithm. We recall that the time complexity of Picard's algorithm in our case is $O(|V||E|\log(\frac{|V|^2}{|E|}))$. Hence the overall running time of our algorithm is $O(|V|^2|E|^2\log(\frac{|V|^2}{|E|}))$. If we denote by n the number of vertices of the input graph, then this running time is $O(n^6\log n)$.

To prove that the maximum-weight convex set problem on a chordal graph can be solved in this running time we need to show that all the information we need in Algorithms 1 and 2 can be computed in a time bounded asymptotically by the time of the preprocessing step. More precisely, given the chordal graph G, we can compute the following information in $O(|V|^2|E|^2\log(\frac{|V|^2}{|E|}))$ time: the clique-separator graph $\mathcal{G}$ of G, the vertices in $G\ominus a$ for each a in $Ar_\mathcal{G}$, the cliques in $\mathcal{K}_G$ for which a is K-blocking for each a in $Ar_\mathcal{G}$, the clique in $\mathcal{K}_a$ for each a in $Ar_\mathcal{G}$, the matrix of the relations $\leqslant_K$ for each K in $\mathcal{K}_G$ and a total order on the arcs $\mathcal{G}$ such that $G\ominus a\subseteq G\ominus a'$ implies $a<a'$ for a, a' in $Ar_\mathcal{G}$. The detailed proofs are given below. This concludes the proof of Theorem 3.

5.1 Detailed time complexity

We now prove that for a chordal graph $G=(V,E)$ given by its adjacency matrix, and a weight function on the vertices, the running time of our algorithm is in $O(|V|^2|E|^2\log(\frac{|V|^2}{|E|}))$, in the worst-case. We need to check that the information needed for the execution of Algorithms 1 and 2 can be computed in advance, only once for a given graph. We recall that for a graph G, finding a connected component of $G-X$ that contains v where $X\subseteq V$ and $v\in V\setminus X$ can be done in time $O(|V|+|E|)$ as stated by Hopcroft and Tarjan [23]. We will use the following lemma.

Lemma 8 *Given a (connected) chordal graph $G=(V,E)$ with clique-separator graph $\mathcal{G}$, we have $|V|-1\leqslant|E|$ and $|Ar_\mathcal{G}|\leqslant|E|$.*

Proof: The first inequality follows from connectedness. For the second inequality, notice each arc $(S_1,S_2)\in Ar_\mathcal{G}$ is generated by two cliques S_1 and S_2 such that $S_1\subset S_2$ and there is no vertex S_3 such that $S_1\subset S_3\subset S_2$. We assign to each arc $a=(S_1,S_2)$ a unique edge $\{s_1,s_2\}$ in G with $s_1\in S_1$ and $s_2\in S_2$. This implies $|Ar_\mathcal{G}|\leqslant|E|$. $\square$

Lemma 9 *Given a chordal graph $G=(V,E)$ and its clique-separator graph $\mathcal{G}$, listing $G\ominus a$ for all a in $Ar_\mathcal{G}$ can be done in $O(|E|^2)$.*

Proof: For each arc $a=(S_1,S_2)$, we need to find the connected component of $G-S_1$ that intersects S_2. Using Lemma 8, we have $O(|E|^2)$ as total running time. $\square$

Lemma 10 *Given a chordal graph $G=(V,E)$ with clique-separator graph $\mathcal{G}$, obtaining the list, for each a in $Ar_\mathcal{G}$, of the cliques in $\mathcal{K}_G$ for which a is K-blocking takes $O(|E|^2|V|^2)$.*

Proof: By Lemma 9 we can obtain the list of $G \ominus a$ for all a in $Ar_{\mathcal{G}}$ in $O(|E|^2)$. Then, for each $a = (S_1, S_2)$ in $Ar_{\mathcal{G}}$, and for each K in $\mathcal{K}_G$, we check two conditions. First, that d_K is not in $G \ominus a$ (i.e. S_1 is a $d_K s_2$-separator for s_2 in S_2). Second, that there is no $a' = (S_3, S_1)$ in $Ar_{\mathcal{G}}$ such that d_K is not in $G \ominus a'$ (i.e. S_1 is minimal among the $s_2 d_K$-separator for every s_2 in S_2). $\square$

Lemma 11 *Given a chordal graph $G = (V, E)$ and its clique-separator graph $\mathcal{G}$, obtaining the relations $\leqslant_K$ for all K in $\mathcal{K}_G$ takes time $O(|V|^2|E|^2)$.*

Proof: By Lemmas 9 and 10 we can obtain, for each arc a in $Ar_{\mathcal{G}}$, the vertices in $G \ominus a$ and the cliques in $\mathcal{K}_G$ for which a is K-blocking in $Ar_{\mathcal{G}}$ in $O(|E|^2|V|^2)$. First, for all K in $\mathcal{K}_G$, we set $d_k \leqslant_K k$ for all k in K. Second, for all K in $\mathcal{K}_G$, and for all $a = (S_1, S_2)$ in $Ar_{\mathcal{G}}$ such that a is K-blocking, we set $s \leqslant_K u$ for all $s \in S_1$ and $u \in (G \ominus a) - S_1$. $\square$

Lemma 12 *Given a chordal graph $G = (V, E)$ and its clique-separator graph $\mathcal{G}$, sorting the arcs of $Ar_{\mathcal{G}}$ such that $G \ominus a \subseteq G \ominus a'$ implies $a < a'$ can be done in $O(|V||E|^2)$ time.*

Proof: We use Lemma 9 to obtain a list of $G \ominus a$ for all a in $Ar_{\mathcal{G}}$. The comparison between $G \ominus a$ and $G \ominus a'$ for a, a' in $Ar_{\mathcal{G}}$ takes $O(|V|)$ time. Hence sorting takes $O(|V||E|^2)$ time. $\square$

Lemma 13 *Given a chordal graph $G = (V, E)$ with clique-separator graph $\mathcal{G}$ and the list of $G \ominus a$ for all a in $Ar_{\mathcal{G}}$, obtaining elements in $\mathcal{K}_a$ for all a in $Ar_{\mathcal{G}}$ takes $O(|E||V|^2)$ time.*

Proof: For each arc $a = (S_1, S_2)$ in $Ar_{\mathcal{G}}$, we look at each $K \in \mathcal{K}_G$ such that $S_1 \subset K$, and we check if the clique is in $G \ominus a$. $\square$
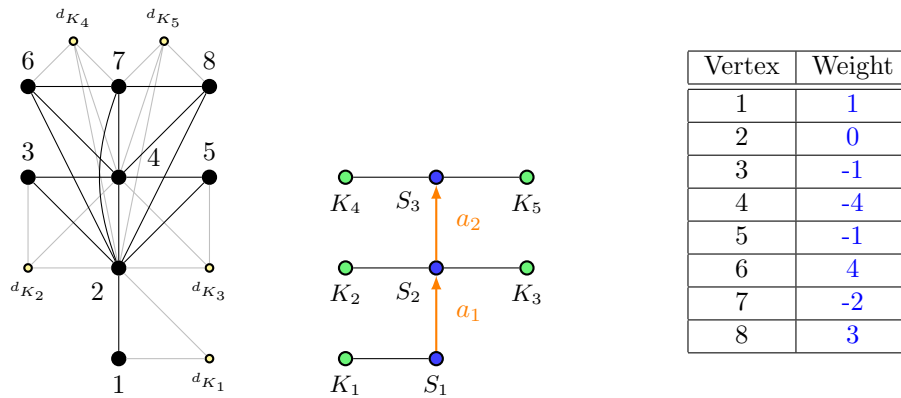

Figure 8: A chordal graph and its clique-separator graph for illustrating Algorithms 1 and 2.
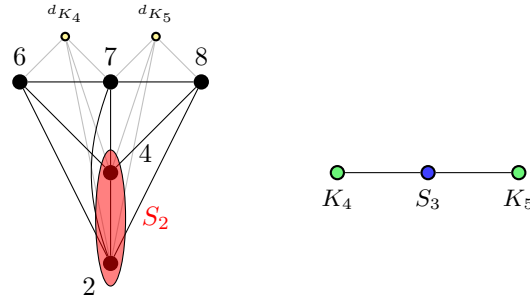
Figure 9: The graph $G \ominus a_2$ and its clique-separator graph.

6   Example

Looking at Algorithms 1 and 2, it seems possible to merge them to save computation time. But the situation is not that simple, because in Theorem 6, the assumption that a is K-blocking cannot be removed. To illustrate the mechanism of the two algorithms, we give an example. Figure 8 shows a chordal graph G, its clique-separator graph $\mathcal{G}$ and a weight function. The goal is to compute $\mathrm{opt}(G, d_{K_1})$.

We look at the preprocessing phase first, and we use Algorithm 2 to label a_1 and a_2. Because $G \ominus a_2 \subset G \ominus a_1$, we begin by computing label(a_2). In other words, we want to compute a maximum-weight S_2-rooted convex set of $G \ominus a_2$, illustrated in Figure 9. There is no K-blocking arc in $G \ominus a_2$ for any K in $\mathcal{K}_{a_2}$, we can directly use Picard's algorithm on each clique in $\mathcal{K}_{a_2}$ after temporarily changing the weight of the vertices in S_2 in order to impose that S_2 be contained in the solution. So Picard's algorithm is used two times, with K_4 and K_5, and we keep the best solution among the outputs. The result will be a convex set of weight 1, for instance $\{2, 4, 6, 7, 8, d_{K_5}\}$, which becomes label$(a_2)$.

Now we compute label(a_1), so we are looking for an S_1-rooted convex set in $G \ominus a_1$ represented in Figure 10. We temporarily change the weight of the vertices in S_1 and we run Algorithm 1 on the graph $G \ominus a_1$ four times, with the cliques K_2, K_3, K_4 and K_5. For the clique K_4 there is no K_4-blocking arc and we use Picard's algorithm. The same process is applied with K_5. For the clique K_2, the arc a_2 is K_2-blocking but already labeled. So we identify the vertices of $(G \ominus a_2) - S_2$ to a vertex z_{a_2} with weight $w(\mathrm{label}(a_2)) - w(S_2) = 1$. After this operation, there is no K_2-blocking arc and we use Picard's algorithm. Figure 11 shows a visual representation of the transformation. The same process is applied with K_3. For the labeling of a_1 we have used Picard's algorithm four times. A best S_1-rooted convex set of $G \ominus a_1$ is $\{2, 6, d_{K_4}\}$ with weight 4.

Now every arc is labeled, and we look at the computing phase. Because we want a maximum-weight d_{K_1}-rooted convex set, we run Algorithm 1. There is only one K_1-blocking arc, namely a_1. So we identify the vertices of $(G \ominus a_1) - S_1$ to a vertex z_{a_1} with weight $w(\mathrm{label}(a_1)) - w(S_1) = 4$. After this operation, there
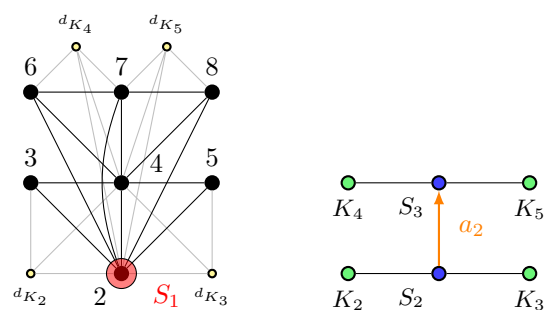
Figure 10: The graph $G \ominus a_1$ and its clique-separator graph.

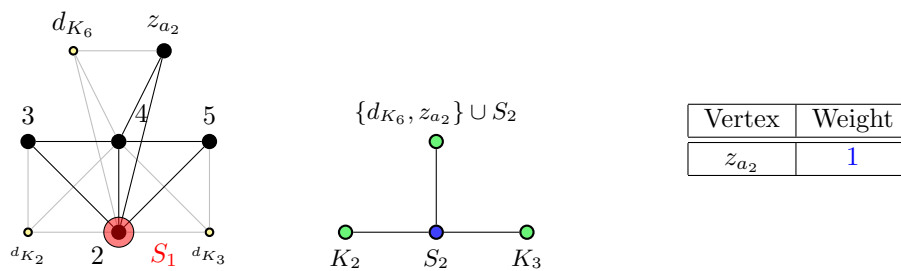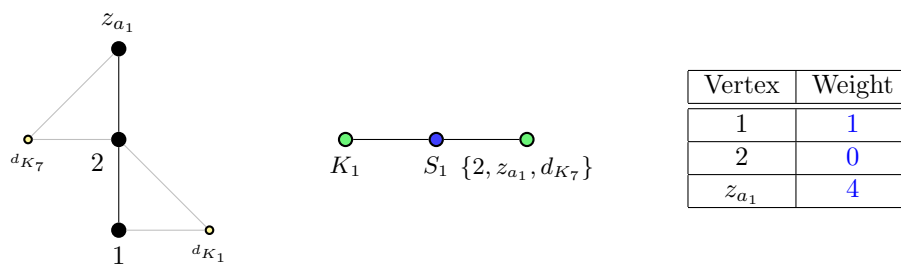Figure 11: The graph after the identification of vertices in order to remove K_2-blocking arcs.

Figure 12: The state of the graph after removing the K_1-blocking arc.

is no K_1-blocking arc as shown in Figure 12, and we use Picard's algorithm to find $\{1, 2, z_{a_1}, d_{K_1}\}$ as a maximum-weight K_1-rooted convex set, which gives rise to the convex set $C^* = \{1, 2, d_{K_1}\} \cup (\text{label}(a_1) \setminus S_1) = \{1, 2, 6, d_{K_1}, d_{K_4}\}$ of G, with $w(C^*) = 5$.

7 Conclusion

We described and studied an algorithm solving the maximum-weight convex set problem in chordal graphs in polynomial time. Our approach relies on structural results of chordal graphs (given by the clique-separator graph) and an efficient algorithm that finds a maximum-weight ideal in a poset (Picard's algorithm). We hope our paper will pave the way for further research on other maximum-weight convex set problems, for instance in convex geometries that arise from AT-free graphs (see Chang *et al.* [6]). It would be interesting to better assess the frontier between convex geometries for which the maximum-weight convex set problem is solvable in polynomial time, and the other convex geometries. Finally, we would like to thank both reviewers for their insightful comments.

References

[1] E. H. Aghezzaf, T. L. Magnanti, and L. A. Wolsey. Optimizing constrained subtrees of trees. *Math. Programming*, 71(2, Ser. A):113–126, 1995. `doi: 10.1007/BF01585993`.

[2] E. A. Algaba, J. M. Bilbao, R. van den Brink, and A. Jiménez-Losada. Cooperative games on antimatroids. *Discrete Mathematics*, 282(1-3):1–15, 2004. `doi:10.1016/j.disc.2003.10.019`.

[3] J. R. S. Blair and B. Peyton. An introduction to chordal graphs and clique trees. In *Graph theory and sparse matrix computation*, volume 56 of *IMA Vol. Math. Appl.*, pages 1–29. Springer, New York, 1993. `doi: 10.1007/978-1-4613-8369-7_1`.

[4] E. A. Boyd and U. Faigle. An algorithmic characterization of antimatroids. *Discrete Appl. Math.*, 28(3):197–205, 1990. `doi:10.1016/0166-218X(90) 90002-T`.

[5] J. Cardinal, J.-P. Doignon, and K. Merckx. On the shelling antimatroids of split graphs. *Discrete Mathematics & Theoretical Computer Science*, Vol 19 no. 1, Mar. 2017. `doi:10.23638/DMTCS-19-1-7`.

[6] J.-M. Chang, T. Kloks, and H.-L. Wang. Gray codes for AT-free orders via antimatroids. In *Combinatorial algorithms*, volume 9538 of *Lecture Notes in Comput. Sci.*, pages 77–87. Springer, [Cham], 2016. `doi:10. 1007/978-3-319-29516-9_7`.

[7] M. Changat and J. Mathew. Induced path transit function, monotone and Peano axioms. *Discrete Math.*, 286(3):185–194, 2004. `doi:10.1016/j. disc.2004.02.017`.

[8] R. Diestel. *Graph theory*, volume 173 of *Graduate Texts in Mathematics*. Springer, Heidelberg, fourth edition, 2010. `doi:10.1007/ 978-3-642-14279-6`.

[9] R. P. Dilworth. Lattices with unique irreducible decompositions. *Ann. of Math. (2)*, 41:771–777, 1940.

[10] M. C. Dourado, D. Rautenbach, and P. M. Schäfer. On finite convexity spaces induced by sets of paths in graphs. *Discrete Math.*, 311(8-9):616–619, 2011. `doi:10.1016/j.disc.2010.12.024`.

[11] P. H. Edelman and R. E. Jamison. The theory of convex geometries. *Geom. Dedicata*, 19(3):247–270, 1985. `doi:10.1007/BF00149365`.

[12] D. Eppstein. Pruning antimatroids is hard, 2007. URL: `https: //11011110.github.io/blog/2007/02/18/pruning-antimatroids-is. html`.

[13] D. Eppstein. The parametric closure problem. *ACM Trans. Algorithms*, 14(1):2:1–2:22, 2018. `doi:10.1145/3147212`.

[14] D. Eppstein, M. Overmars, G. Rote, and G. Woeginger. Finding minimum area *k*-gons. *Discrete Comput. Geom.*, 7(1):45–58, 1992. `doi:10.1007/BF02187823`.

[15] J.-C. Falmagne and J.-P. Doignon. *Learning Spaces*. Springer-Verlag, Berlin, 2011.

[16] M. Farber and R. E. Jamison. Convexity in graphs and hypergraphs. *SIAM J. Algebraic Discrete Methods*, 7(3):433–444, 1986. `doi:10.1137/0607049`.

[17] M. Farber and R. E. Jamison. On local convexity in graphs. *Discrete Math.*, 66(3):231–247, 1987. `doi:10.1016/0012-365X(87)90099-9`.

[18] D. R. Fulkerson and O. A. Gross. Incidence matrices and interval graphs. *Pacific J. Math.*, 15(3):835–855, 1965. URL: `http://projecteuclid.org/euclid.pjm/1102995572`.

[19] A. V. Goldberg and R. E. Tarjan. A new approach to the maximum-flow problem. *J. Assoc. Comput. Mach.*, 35(4):921–940, 1988. `doi:10.1145/48014.61051`.

[20] M. C. Golumbic. *Algorithmic graph theory and perfect graphs*, volume 57 of *Annals of Discrete Mathematics*. Elsevier Science B.V., Amsterdam, second edition, 2004. With a foreword by Claude Berge.

[21] H. Gröflin. Path-closed sets. *Combinatorica*, 4(4):281–290, 1984. `doi:10.1007/BF02579138`.

[22] C. W. Ho and R. C. T. Lee. Counting clique trees and computing perfect elimination schemes in parallel. *Inform. Process. Lett.*, 31(2):61–68, 1989. `doi:10.1016/0020-0190(89)90070-7`.

[23] J. Hopcroft and R. Tarjan. Algorithm 447: Efficient algorithms for graph manipulation. *Commun. ACM*, 16(6):372–378, June 1973. `doi:10.1145/362248.362272`.

[24] L. Ibarra. The clique-separator graph for chordal graphs. *Discrete Appl. Math.*, 157(8):1737–1749, 2009. `doi:10.1016/j.dam.2009.02.006`.

[25] R. E. Jamison-Waldner. Partition numbers for trees and ordered sets. *Pacific J. Math.*, 96(1):115–140, 1981. URL: `http://projecteuclid.org/euclid.pjm/1102734951`.

[26] R. E. Jamison-Waldner. A perspective on abstract convexity: classifying alignments by varieties. In *Convexity and related combinatorial geometry (Norman, Okla., 1980)*, volume 76 of *Lecture Notes in Pure and Appl. Math.*, pages 113–150. Dekker, New York, 1982.

[27] B. Kannan and M. Changat. Hull numbers of path convexities on graphs. In *Convexity in discrete structures*, volume 5 of *Ramanujan Math. Soc. Lect. Notes Ser.*, pages 11–23. Ramanujan Math. Soc., Mysore, 2008.

[28] D. M. King, S. H. Jacobson, E. C. Sewell, and W. K. T. Cho. Geo-graphs: An efficient model for enforcing contiguity and hole constraints in planar graph partitioning. *Operations Research*, 60(5):1213–1228, 2012. `doi:10.1287/opre.1120.1083`.

[29] B. Korte and L. Lovász. Polyhedral results for antimatroids. *Annals New York Academy of Sciences*, 555:283–295, 1989. `doi:10.1111/j.1749-6632.1989.tb22462.x`.

[30] B. Korte, L. Lovász, and R. Schrader. *Greedoids*, volume 4 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin, 1991. `doi:10.1007/978-3-642-58191-5`.

[31] G. A. Koshevoy. Choice functions and abstract convex geometries. *Math. Social Sci.*, 38(1):35–44, 1999. `doi:10.1016/S0165-4896(98)00044-4`.

[32] T. A. McKee and F. R. McMorris. *Topics in intersection graph theory*. SIAM Monographs on Discrete Mathematics and Applications. SIAM, 1999. `doi:10.1137/1.9780898719802`.

[33] D. Orlin. Optimal weapons allocation against layered defenses. *Naval Research Logistics (NRL)*, 34(5):605–617, 1987. `doi:10.1002/1520-6750(198710)34:5<605::AID-NAV3220340502>3.0.CO;2-L`.

[34] J. Oxley. *Matroid theory*, volume 21 of *Oxford Graduate Texts in Mathematics*. Oxford University Press, Oxford, second edition, 2011. `doi:10.1093/acprof:oso/9780198566946.001.0001`.

[35] A. F. Parkinson. *Essays on sequence optimization in block cave mining and inventory policies with two delivery sizes*. PhD thesis, University of British Columbia, 2012. `doi:10.14288/1.0073023`.

[36] H. P. Patil. On the structure of k-trees. *J. Combin. Inform. System Sci.*, 11(2-4):57–64, 1986.

[37] J.-C. Picard. Maximal closure of a graph and applications to combinatorial problems. *Management Sci.*, 22(11):1268–1272, 1975/76.

[38] J. M. W. Rhys. A selection problem of shared fixed costs and network flows. *Management Science*, 17(3):200–207, 1970. `doi:10.1287/mnsc.17.3.200`.

[39] J. B. Sidney. Decomposition algorithms for single-machine sequencing with precedence relations and deferral costs. *Operations Research*, 23(2):283–298, 1975. URL: `http://www.jstor.org/stable/169530`.

[40] J. C. Williams. Convex land acquisition with zero-one programming. *Environment and Planning B: Planning and Design*, 30(2):255–270, 2003. `doi:10.1068/b12925`.