Bond Polytope under Vertex- and Edge-sums

Petr Kolman and Hans Raj Tiwary

Charles University, Faculty of Mathematics and Physics
Department of Applied Mathematics
Prague, Czech Republic
{kolman,hansraj}@kam.mff.cuni.cz

Abstract. A cut in a graph G is called a *bond* if both parts of the cut induce connected subgraphs in G, and the *bond polytope* is the convex hull of all bonds. Computing the maximum weight bond is an NP-hard problem even for planar graphs. However the problem is solvable in linear time on $(K_5 \setminus e)$-minor-free graphs, and in more general, on graphs of bounded treewidth, essentially due to clique-sum decomposition into simpler graphs.

We show how to obtain the bond polytope of graphs that are 1- and 2-sum of graphs G_1 and G_2 from the bond polytopes of G_1, G_2. Using this we show that the extension complexity of the bond polytope of $(K_5 \setminus e)$-minor-free graphs is linear. Prior to this work, a linear size description of the bond polytope was known only for 3-connected planar $(K_5 \setminus e)$-minor-free graphs, essentially only for wheel graphs.

We also describe an elementary linear time algorithm for the Max-Bond problem on $(K_5 \setminus e)$-minor-free graphs. Prior to this work, a linear time algorithm in this setting the hidden constant in the big-Oh notation was huge due to the fact that the known algorithm uses the heavy machinery of linear time algorithms for graphs of bounded treewidth, as a black box.

1 Introduction

The Max-Cut problem is a fundamental problem in computer science and is one of Karp's original 21 NP-Complete problems [17]. Given a graph $G = (V, E)$ the problem asks for a subset $S \subseteq V$ of vertices such that the number of edges with exactly one endpoint in S is as large as possible. However, in some applications such as image segmentation [24], forest planning and harvest scheduling [4], and certain market zoning [13], one imposes an additional condition that both components $G[S]$ and $G[V \setminus S]$ be *connected.* This version of Max-Cut has been studied by various authors [10,8,11,14,6,5] under different names, but following Duarte et al. [9] and Chimani et al. [6] we will refer to this as the *bond problem.*

Formally, given a graph $G = (V, E)$ a *bond* in G is (the set of edges in) a cut $(S, V \setminus S)$ such that the induced subgraphs $G[S]$ and $G[V \setminus S]$ are both connected. The Max-Bond problem seeks to find a bond $(S, V \setminus S)$ such that the number of edges between S and $V \setminus S$ is maximized. For each bond in a graph G, we consider the characteristic vector of its edges; the convex hull of all such vectors defines the *bond polytope* of G. For simplicity we do not distinguish between a bond, the edges in a bond, and the characteristic vector of the edges in a bond unless the meaning is not clear in the context of discussion.

In this paper we deal with the MAX-BOND problem on $(K_5 \setminus e)$-minor-free graphs. For this class of graphs, Chaourar [5] gave a quadratic time algorithm for finding a maximum bond. Chimani et al. [6] improved this result by giving a linear time algorithm; the algorithm uses as a black box a linear time algorithm of Duarte et al. [9] for the bond problem on bounded tree-width graphs, and this black-box is used to get the maximum bond for the wheel graphs W_n. Chimani et al. [6] also gave characterisation of the bond polytope for 3-connected planar $(K_5 \setminus e)$-minor-free graphs by giving a linear size set of linear inequalities defining it; by a result of Wagner [25], this class of graphs contains only the wheel graphs W_n, the triangle K_3, and the triangular prism. The question of describing the bond polytope for general $(K_5 \setminus e)$-minor-free graphs was left open.

Our contributions are twofold:

1. We show how to obtain linear size descriptions of the bond polytope of graphs that are k-sum, for $k = 1, 2$, of other graphs with known linear size bond polytope. Using this result, we prove that the extension complexity of the bond polytope is linear for arbitrary $(K_5 \setminus e)$-minor-free graphs. This, in a sense, is the best one can do because – as we note later – the actual description of the bond polytope even for $(K_5 \setminus e)$-minor-free graphs can be exponential in the size of the graph. This answers an open question posed by Chimani et al. [6].
2. We simplify the algorithmic result for the MAX-BOND problem of Chimani et al. [6] by giving a simple linear time algorithm for the wheel graph, removing the need to use the tree-width machinery [10], which yields a linear time algorithm for the MAX-BOND problem for all $(K_5 \setminus e)$-minor-free graphs. Chimani et al. [6] mention the possibility of existence of algorithms simpler than theirs so our algorithm can be seen as an answer to their question.

It should be noted that $(K_5 \setminus e)$-minor-free graphs have bounded treewidth and bonds can be represented by a formula in Monadic Second Order (MSO) logic. So both a linear time algorithm as well as a linear size extended formulation follow readily from meta results about bounded treewidth graphs: the algorithmic results follow from the work of Courcelle [7] and are given explicitly by Duarte et al. [9], while the polyhedral results follow from the work of Kolman et al. [18]. However, the magnitude of the constants in both cases is enormous [19], in contrast to the constants in our results.

Other Related Results Cut polytope for clique-sums of size three was studied by Barahona [2] who gave efficient algorithm and extended formulation for Cut polytope of K_5-minor-free graphs.

A closely related problem is the version of the MAX-CUT in which only the S part is required to be connected. This version of the MAX-CUT problem is NP-hard [14] as well. Schieber and Vahidi [22] gave an $O(\log \log n)$-approximation improving an earlier $O(\log n)$-approximation [14].

In contrast to the MAX-CUT problem, there is no constant-factor approximation algorithm for MAX-BOND unless P=NP [8]. On the positive side, both MAX-BOND and the version of MAX-CUT with one side connected are fixed-parameter tractable when parameterized by the size of the solution, the treewidth, and the twin-cover number [8].

2 Preliminaries

Let G_1 and G_2 be two graphs and $U_1 \subseteq V(G_1)$ and $U_2 \subseteq V(G_2)$ two subsets of vertices inducing a clique of the same size, say size k, for some $k \geq 1$. A graph G is a *clique-sum* of G_1 and G_2 if G is obtained from G_1 and G_2 by identifying U_1 and U_2, and possibly removing some edges from the clique.

In this paper, we use the clique-sums for $k = 1, 2$. To distinguish between the 2-sum that keeps the edge in the clique, and the 2-sum that removes it, we denote the former operation by $\oplus_2$ and the later by $\oplus_2^-$. If we want to emphasize that $G_1 \oplus_1 G_2$ is taken over a vertex v, we will denote it as $G_1 \oplus_v G_2$. Similarly, $G_1 \oplus_e G_2$ or $G_1 \oplus_e^- G_2$ will be used to mean that the 2-sum of G_1 and G_2 is taken over the edge e.

For a graph $G = (V, E)$, a pair of vertices uv is called a *non-edge* if $uv \notin E$. For an edge $e \in E$, by $G \setminus e$ we denote the graph $(V, E \setminus \{e\})$, by $G \cup \{e\}$ the graph $(V, E \cup e)$, and we use uv as an abbreviation of $\{u, v\}$. In the case of (edge) weighted graphs, the weight of an edge uv is denoted $w(u, v)$. For a subset S of vertices, $\delta(S)$ is the set of edges between S and $V \setminus S$.

A graph H is a *minor* of a graph G if H can be obtained from G be a series of vertex and edge deletions and edge contractions. A graph G is H*-minor-free* if H is not a minor of G.

For $n \geq 3$, a *wheel graph* W_n is a graph with a vertex set $V = \{0, 1, \ldots, n-1\} \cup \{c\}$, for $c \notin \{0, 1, \ldots, n-1\}$, and an edge set $E = \bigcup_{i=0}^{n-2} \{(i, i+1), (i, c)\} \cup \{(n-1, 0), (n-1, c)\}$; the vertex c is called the *hub* of the wheel, the cycle $\{0, 1, \ldots, n-1, 0\}$ is the *rim*, and the edges of the form (i, c) are the *spokes* of the wheel. For integers $i < j$, let $[i, j]$ denote the set $\{i, i+1, \ldots, j\}$.

Theorem 1 (Satz 7, Wagner [25]). *Each maximal $(K_5 \setminus e)$-minor-free graph G can be decomposed as $G = G_1 \oplus^1 \cdots \oplus^{l-1} G_\ell$ where each G_i is isomorphic to a wheel graph, Prism, K_2, K_3, or $K_{3,3}$, and each operation $\oplus^i$ is $\oplus_1$ or $\oplus_2$.*

Theorem 2. *Each $(K_5 \setminus e)$-minor-free graph G can be decomposed in linear time as $G = G_1 \oplus^1 \cdots \oplus^{l-1} G_\ell$ where each G_i is isomorphic to a wheel graph, Prism, K_2, K_3, or $K_{3,3}$, and each operation $\oplus^i$ is $\oplus_1$, $\oplus_2$ or $\oplus_2^-$.*

The proof of Theorem 2 is in the Appendix.

Let P be a polytope in $\mathbb{R}^d$. A polytope Q in $\mathbb{R}^{d+r}$ is called an *extended formulation* of P if P is a projection of Q onto the first d coordinates. The *size* of a polytope is defined to be the number of its facet-defining inequalities, and the *extension complexity* of a polytope P, denoted by $\mathrm{xc}(P)$, is the size of its smallest extended formulation.

Theorem 3 (Balas [1], Theorem 2.1). *If $P_1, \ldots, P_q$ are non-empty polytopes, then*

$$\mathrm{xc}\left(\mathrm{conv}\Big(\bigcup_{i=1}^{q} P_i\Big)\right) \leq q + \sum_{i=1}^{q} \mathrm{xc}\,(P_i) \ .$$

Furthermore, such an extended formulation can be constructed from extended formulations of the P_i's in linear time.

Let $P_1 \subseteq \mathbb{R}^{d_1+k}$ and $P_2 \subseteq \mathbb{R}^{d_2+k}$ be two 0/1-polytopes with vertices $\mathrm{vert}(P_1)$ and $\mathrm{vert}(P_2)$, respectively. The *glued product* $P_1 \times_k P_2$ of P_1 and P_2, where the gluing is done over the last k coordinates, is defined to be

$$P_1 \times_k P_2 := \mathrm{conv}\left\{ \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{z} \end{pmatrix} \in \{0,1\}^{d_1+d_2+k} \;\middle|\; \begin{pmatrix} \mathbf{x} \\ \mathbf{z} \end{pmatrix} \in \mathrm{vert}(P_1), \begin{pmatrix} \mathbf{y} \\ \mathbf{z} \end{pmatrix} \in \mathrm{vert}(P_2) \right\}.$$

We will use the following known result about glued products.

Lemma 4 (Gluing lemma [20,18]). *Let P and Q be 0/1-polytopes and let the k (glued) coordinates in P be labeled $z_1, \ldots, z_k$, and the k (glued) coordinates in Q be labeled $w_1, \ldots, w_k$. Suppose that $\mathbf{1}^\intercal z \leqslant 1$ is valid for P and $\mathbf{1}^\intercal w \leqslant 1$ is valid for Q. Then $\mathrm{xc}\,(P \times_k Q) \leqslant \mathrm{xc}\,(P) + \mathrm{xc}\,(Q)$.*

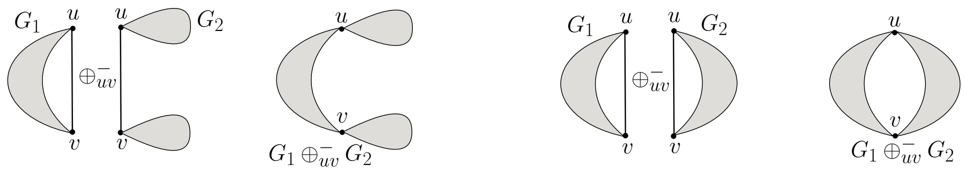
We conclude this section with a lemma about bonds of $G_1 \oplus_{uv}^{-} G_2$; analogous claims about bonds of $G_1 \oplus_{uv} G_2$ and $G_1 \oplus_u G_2$ were observed earlier [5,6].

Lemma 5. *Let G_1, G_2 be graphs, $uv \in E(G_1) \cap E(G_2)$ be an edge that appears in both of them, and let $G = G_1 \oplus_{uv}^{-} G_2$. Then the following claims hold:*

1. *If $G_1 \setminus uv$ is connected but $G_2 \setminus uv$ is not, then $F \subseteq E(G)$ is a bond of G if and only if*
   - *F is a bond of $G_1 \setminus uv$, or*
   - *F is a bond of G_2 with u and v on the same side.*
2. *If both $G_1 \setminus uv$ and $G_2 \setminus uv$ are connected, then $F \subseteq E(G)$ is a bond of G if and only if*
   - *F is a bond of $G_1 \setminus uv$ with u and v on the same side, or*
   - *F is a bond of $G_2 \setminus uv$ with u and v on the same side, or*
   - *$F \cap E(G_i)$ is a bond of $G_i \setminus uv$ with u and v on different sides, for both $i \in \{1,2\}$.*

Proof. The two cases are illustrated in Figures 1a-1b. Case 1. We start with the left to right implication. By the assumption, $G_2 \setminus uv$ has two connected components, one containing the vertex u and the other containing the vertex v; we denote them H_u and H_v. We distinguish two subcases: either i) F is a bond of G with u and v on the same side, or ii) F is a bond of G with u and v on different sides. In the first subcase, either F is a bond of G_1 with u and v on the same side, or F is a bond of G_2 with u and v on the same side.

In the other subcase, F must be a bond of $G_1 \setminus uv$ with u and v on different sides. Combining the two subcases completes the proof of the left-to-right implication of Case 1.



(a) Case 1. (b) Case 2.

Fig. 1: 2-sum

The right-to-left implication in Case 1 is obvious.

Case 2. As in the proof of Case 1, we distinguish several subcases. If F is a bond of G with u and v on the same side, then one of the subgraphs $G_1 \setminus uv$ and $G_2 \setminus uv$ is untouched by the bond F of G, and F is a bond of the other part. If F is a bond of G with u and v on different sides, then the assumption of connectivity of both $G_1 \setminus uv$ and $G_2 \setminus uv$ implies that $F \cap E(G_i)$ is a bond of $G_i \setminus uv$ with u and v on different sides, for $i = 1, 2$.

The right-to-left implication in Case 2 is obvious. □

3 Extension Complexity

Let $G = (V, E)$ be a graph and let $E' \subseteq \binom{V}{2} \setminus E$ be a subset of non-edges of G. An *Augmented Bond Polytope* $\mathrm{BOND}(G, E')$ is the convex hull of vectors $\mathbf{x} \in \mathbb{R}^{|E|+|E'|}$ where $\mathbf{x}_E$ is the characteristic vector of a bond $F \subseteq E$ in G and for every $uv \in E'$, $\mathbf{x}_{uv} = 0$ if u and v are on the same side of the bond F and $\mathbf{x}_{uv} = 1$ uf u and v are on different sides of the bond F.

Lemma 6. *Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two graphs with $V_1 \cap V_2 = \{v\}$, and let $E_1' \subseteq \binom{V_1}{2} \setminus E_1, E_2' \subseteq \binom{V_2}{2} \setminus E_2$. Let $\mathrm{BOND}(G_1, E_1')$, $\mathrm{BOND}(G_2, E_2')$ be their respective augmented bond polytopes. Suppose $\mathrm{BOND}(G_1, E_1') = \mathrm{conv}(\mathbf{B}_1)$ and $\mathrm{BOND}(G_2, E_2') = \mathrm{conv}(\mathbf{B}_2)$. Then,*

$$\mathrm{BOND}(G_1 \oplus_v G_2, E_1' \cup E_2') = \mathrm{conv}\begin{pmatrix} \mathbf{B}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_2 \end{pmatrix}$$

where the right-hand side is a shorthand of $\mathrm{conv}(\{(\mathbf{b}, \mathbf{0}) | \mathbf{b} \in \mathbf{B}_1\} \cup \{(\mathbf{0}, \mathbf{b}) | \mathbf{b} \in \mathbf{B}_2\})$.

Proof. Notice that any cut in $G_1 \oplus_v G_2$ either cuts G_1 in two components placing all of G_2 in the component containing the common vertex v or it cuts G_2 in two components placing all of G_1 in the component containing the common vertex v. Therefore, any bond in $G_1 \oplus_v G_2$ is either a bond in G_1 extended with zeroes at the coordinates x_{uw} for $uw \in E_2 \cup E_2'$, or a bond in G_2 extended with zeroes at the coordinates x_{uw} for $uw \in E_1 \cup E_1'$. □

It should be remarked that the above description of $\mathrm{BOND}(G_1 \oplus_v G_2, E_1' \cup E_2')$ is the subdirect sum of $\mathrm{BOND}(G_1, E_1')$ and $\mathrm{BOND}(G_2, E_2')$ and thus it can have a number of inequalities that is asymptotically the product of the number of inequalities describing the two multiplicands (cf. Lemma 15 in the Appendix). So, in general, one cannot obtain a linear size description for $(K_5 \setminus e)$-minor-free graphs unless one is willing to consider extended formulations.

Lemma 7. *Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two graphs with $E_1 \cap E_2 = \{e\}$, and let $E_1' \subseteq \binom{V_1}{2} \setminus E_1, E_2' \subseteq \binom{V_2}{2} \setminus E_2$. Let $\mathrm{BOND}\,(G_1, E_1')$, $\mathrm{BOND}\,(G_2, E_2')$ be their respective augmented bond polytopes. Suppose $\mathrm{BOND}\,(G_1, E_1') = \mathrm{conv}\,(\mathbf{B}_1)$ and $\mathrm{BOND}\,(G_2, E_2') = \mathrm{conv}\,(\mathbf{B}_2)$. Then, $\mathrm{BOND}\,(G_1 \oplus_e G_2, E_1' \cup E_2')$ is affinely equivalent to*

$$\mathrm{conv}\left\{ \begin{pmatrix} \mathbf{B}_1^0 & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_2^0 \end{pmatrix} \cup (\mathbf{B}_1^1 \times \mathbf{B}_2^1) \right\},$$

where $\mathbf{B}_j^i = \{\mathbf{b} \in \mathbf{B}_j \mid \mathbf{b}_e = i\}$ for $i \in \{0,1\}$ and $j \in \{1,2\}$ and $\times$ denotes the Cartesian product.

Proof. Let $e = \{u, v\}$. Any bond in $G = G_1 \oplus_e G_2$ either has u, v in the same component or in different components. If u, v are in the same component, then the bond is obtained either from a bond of G_1 by putting G_2 entirely in the component containing u, v or from a bond of G_2 by putting G_1 entirely in the component containing u, v. If u, v are in different components, then the bond is obtained from a bond $\mathbf{b}_1$ of G_1 and a bond $\mathbf{b}_2$ of G_2 such that u, v are in different components of both $\mathbf{b}_1$ and $\mathbf{b}_2$. ☐

Theorem 8. *Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two graphs and let $E_1' \subseteq \binom{V_1}{2} \setminus E_1, E_2' \subseteq \binom{V_2}{2} \setminus E_2$. Let $\mathrm{BOND}\,(G_1, E_1')$, $\mathrm{BOND}\,(G_2, E_2')$ be their respective augmented bond polytopes. Suppose $\mathrm{BOND}\,(G_1, E_1') = \mathrm{conv}\,(\mathbf{B}_1)$ and $\mathrm{BOND}\,(G_2, E_2') = \mathrm{conv}\,(\mathbf{B}_2)$. Then,*

$$\mathrm{xc}\left(\mathrm{BOND}\left(G_1 \oplus_k G_2, E_1' \cup E_2'\right)\right) \leqslant \mathrm{xc}\left(\mathrm{BOND}\left(G_1, E_1'\right)\right) + \mathrm{xc}\left(\mathrm{BOND}\left(G_2, E_2'\right)\right) + \mathcal{O}\,(1)\,,$$

for $k \in \{1, 2\}$. Furthermore, given extended formulations for $\mathrm{BOND}\,(G_1, E_1')$ and $\mathrm{BOND}\,(G_2, E_2')$, an extended formulation for $\mathrm{BOND}\,(G_1 \oplus_k G_2, E_1' \cup E_2')$ can be constructed in linear time.

Proof. For $k = 1$, the result is an immediate corollary of Lemma 6 and Theorem 3. For $k = 2$, let e be the edge along which the 2-sum is taken, and let d be dimension of the affine hull of $\mathrm{BOND}\,(G_1, E_1')$.

First, we assume that $\mathrm{BOND}\,(G_1, E_1')$ is embedded in $\mathbb{R}^{d+2}$. Call the extra two coordinates w, z and the embedded polytope P_{G_1}. We assume that the following property holds for each $\mathbf{b} \in \mathrm{vert}\,(P_{G_1})$:

$$\begin{aligned} (\mathbf{b})_e = 0 &\implies (\mathbf{b})_w = 0, (\mathbf{b})_z = 1 \\ (\mathbf{b})_e = 1 &\implies (\mathbf{b})_w = 1, (\mathbf{b})_z = 1 \end{aligned}$$

This can be achieved by taking the glued product of $\mathrm{BOND}\,(G_1, E_1')$ with the segment $S = \mathrm{conv}\,(\{(0,0,1), (1,1,1)\})$ glueing coordinate $(\mathbf{x})_e$ in $\mathrm{BOND}\,(G_1, E_1')$ with the first coordinate of the segment S. This results in an additive $\mathcal{O}\,(1)$ increase in the extension complexity due to Lemma 4. Next we take the convex hull of the union of the resulting polytope and the point $(\mathbf{0}, 1, 0) \in \mathbb{R}^{\mathrm{d}+2}$ to obtain P_{G_1}', again resulting in an $\mathcal{O}\,(1)$ additive increase in the extension complexity due to Theorem 3. We have $\mathrm{xc}\left(P_{G_1}'\right) \leqslant \mathrm{xc}\,(\mathrm{BOND}\,(G_1, E_1')) + \mathcal{O}\,(1)\,.$

Similarly, we obtain P'_{G_2} from $\mathrm{BOND}\left(G_2, E'_2\right)$ by adding new coordinates w, z first ensuring

$$(\mathbf{b})_e = 0 \implies (\mathbf{b})_w = 1, (\mathbf{b})_z = 0$$
$$(\mathbf{b})_e = 1 \implies (\mathbf{b})_w = 1, (\mathbf{b})_z = 1$$

for each $\mathbf{b} \in \mathrm{vert}\left(P_{G_2}\right)$, and then taking the convex hull of the union of the resulting polytope with the point $(\mathbf{0}, 0, 1)$. By the same arguments as for P'_{G_1} we have that $\mathrm{xc}\left(P'_{G_2}\right) \leqslant \mathrm{xc}\left(\mathrm{BOND}\left(G_2, E'_2\right)\right) + \mathcal{O}\left(1\right)$.

Finally, we take the glued-product of P'_{G_1} and P'_{G_2} where the gluing is done over the z, w coordinates in P'_{G_1} with the z, w coordinates in P'_{G_2}. The resulting polytope is an extended formulation of $\mathrm{BOND}\left(G_1 \oplus_2 G_2, E'_1 \cup E'_2\right)$ by Lemma 7, and by Theorem 3, it has extension complexity at most $\mathrm{xc}\left(P'_{G_1}\right) + \mathrm{xc}\left(P'_{G_2}\right) + \mathcal{O}\left(1\right)$ which is $\mathrm{xc}\left(\mathrm{BOND}\left(G_1, E'_1\right)\right) + \mathrm{xc}\left(\mathrm{BOND}\left(G_2, E'_2\right)\right) + \mathcal{O}\left(1\right)$. Note that all the steps in the proof are efficiently constructive so the resulting extended formulation can be constructed in linear time. $\square$

Lemma 9. *Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two graphs such that $u, v \in V_1 \cap V_2$ and $\{u, v\} \in E_1 \cap E_2$. Let $E'_1 \subseteq \binom{V_1}{2} \setminus E_1$ and $E'_2 \subseteq \binom{V_2}{2} \setminus E_2$. Suppose that u and v belong to the same connected component in $G_1 \setminus uv$.*

1. *If $G_2 \setminus uv$ is disconnected, then*
$$\mathrm{xc}\left(\mathrm{BOND}\left(G_1 \oplus_{uv}^{-} G_2, E'_1 \cup E'_2 \cup \{uv\}\right)\right) \leqslant \mathrm{xc}\left(\mathrm{BOND}\left(G_1 \setminus uv, E'_1 \cup \{uv\}\right)\right) + \mathrm{xc}\left(\mathrm{BOND}\left(G_2, E'_2\right)\right) + \mathcal{O}\left(1\right).$$
2. *If $G_2 \setminus uv$ is connected, then*
$$\mathrm{xc}\left(\mathrm{BOND}\left(G_1 \oplus_{uv}^{-} G_2, E'_1 \cup E'_2 \cup \{u, v\}\right)\right) \leqslant \mathrm{xc}\left(\mathrm{BOND}\left(G_1 \setminus uv, E'_1 \cup \{uv\}\right)\right) + \mathrm{xc}\left(\mathrm{BOND}\left(G_2 \setminus uv, E'_2 \cup \{uv\}\right)\right) + \mathcal{O}\left(1\right).$$

Furthermore, in each of these cases the resulting extended formulation can be constructed in linear time given extended formulations for appropriate polytopes.

Proof. The claim of Part 1 follows immediately from the characterization in Lemma 5 (Case 1) and Theorem 3. For Part 2, we note the characterization in Lemma 5 (Case 2) and observe that the case is identical to that in Theorem 8 and hence an identical proof yields the result. $\square$

Lemma 10. *Let $G = (V, E)$ be a graph and let $E' \subseteq \binom{V}{2} \setminus E$ such that $G' = (V, E \cup E')$ is a wheel graph. Then, $\mathrm{xc}\left(\mathrm{BOND}\left(G, E'\right)\right) \leqslant \mathcal{O}\left(|V|\right)$. Furthermore, such an extended formulation can be constructed in time $\mathcal{O}\left(|V|\right)$.*

Proof. We prove the lemma by induction on the number of vertices on the rim. For any constant n we have a constant size graph and so it has a constant number of bonds and the extension complexity of any augmented bond polytope is a constant.

Suppose the claim holds for n. That is, for any subgraph of wheel W_n with center c and rim vertices $0, \ldots, n$ the complexity of the corresponding augmented bond polytope (with the non-edges as augmented coordinates) is $\mathcal{O}\left(n\right)$. Now consider a subgraph G_{n+1} of W_{n+1} with a set of non-edges $\overline{E}_{n+1}$.

Either there exists a rim-vertex with degree strictly less than three, or the given graph is the wheel itself and there are no augmented coordinates and thus the extension complexity of BOND $\left(G, \overline{E}_{n+1}\right)$ is linear and can be explicitly described [6]. Similarly, either there exists a rim-vertex with degree strictly more than one, or the given graph is a star and the augmented coordinates correspond to the cycle $0, \ldots, n+1$. The star has only a linear number of bonds - one for each ray being cut off. So an extended formulation can be constructed in linear time using Theorem 3 using each bond explicitly.

Therefore, without loss of generality we can assume that there is a rim vertex that has degree exactly two. For simplicity we assume that this vertex is labeled $n+1$ even though for the construction of an extended formulation the actual label of such a vertex can be directly used. We distinguish following cases:

1. edges $\{n, n+1\}, \{c, n+1\}$ are present while the edge $\{0, n+1\}$ is absent,
2. edges $\{0, n+1\}, \{c, n+1\}$ are present while the edge $\{n, n+1\}$ is absent,
3. edges $\{0, n+1\}, \{n, n+1\}$ are present while the edge $\{c, n+1\}$ is absent.

Cases 1 and 2 are identical except for vertex labeling so we will consider only case 1. We construct a subgraph G_n of W_n as follows. We remove vertex $n+1$. For each $i \in \{0, n\}$, we keep $\{c, i\}$ as an edge in G_n if it was an edge in G_{n+1}, otherwise we keep it as a non-edge in $\overline{E}_n$. Similarly for $\{i, i+1\}$, for $i \in \{0, n-1\}$. Finally, $\{0, n\} \in \overline{E}_n$. By our inductive hypothesis the augmented bond polytope BOND $\left(G_n, \overline{E}_n\right)$ has extension complexity $\mathcal{O}(n)$.

Note that any bond in G_{n+1} must have vertex $n+1$ in the same component as either c or n. The only exception is the bond that has vertex $n+1$ as one component and the rest of the graph in the other component. Since, every unexceptional bond bond in G_{n+1} corresponds to a bond in G_n and there are only finitely many types of bonds – in how they appear at the vertices $c, n, n+1$, and 0, we glue extra coordinates onto BOND $\left(G_n, \overline{E}_n\right)$ encoding this. More specifically, consider the glued product of BOND $\left(G_n, \overline{E}_n\right)$ over the coordinates $x_{c,n}, x_{c,0}, x_{0,n}$ with polytope 3.1. Finally we add the single exceptional bond and the resulting polytope is an extended formulation for BOND $\left(G_{n+1}, \overline{E}_{n+1}\right)$.

Similarly for case 3, we obtain G_n by keeping edges/non-edges as they are in G_{n+1} over vertices $c, 0, \ldots, n$ and $\{0, n\} \in E(G_n)$. Similar to previous case we glue extra coordinates to extend bonds in G_n to bonds in G_{n+1}. We see that the polytope needed for glueing is polytope 3.2.

$$\begin{pmatrix} x_{c,n} & x_{0,n} & x_{c,0} & x_{n,n+1} & x_{n+1,0} & x_{c,n+1} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 \end{pmatrix} \quad \begin{pmatrix} x_{c,n} & x_{0,n} & x_{c,0} & x_{n,n+1} & x_{n+1,0} & x_{c,n+1} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

Polytope 3.1: Vertices

Polytope 3.2: Vertices

Finally we add the single exceptional bond and the resulting polytope is an extended formulation for BOND $\left(G_{n+1}, \overline{E}_{n+1}\right)$.

Therefore we see that $\mathrm{xc}\left(\mathrm{BOND}\left(G_{n+1}, \overline{E}_{n+1}\right)\right) \leqslant \mathrm{xc}\left(\mathrm{BOND}\left(G_n, \overline{E}_n\right)\right) + \mathcal{O}(1)$. Furthermore, observe that a degree two vertex in a subgraph of wheel can be found in linear time so the inductive step can be performed with a constant overhead. This conludes the proof of the inductive step. □

Theorem 11. *Let $G = (V, E)$ be a graph and let $E' \subseteq \binom{V}{2} \setminus E$ such that $G' = (V, E \cup E')$ does not contain $(K_5 \setminus e)$ as a minor. Then,* $\mathrm{xc}(\mathrm{BOND}(G, E')) \leqslant \mathcal{O}(|V|)$. *Furthermore, such an extended formulation can be constructed in time $\mathcal{O}(|V|)$.*

Proof. If G is not connected then depending on whether it has two or more connected components, there is either only the trivial bond or no bonds. Therefore we may assume that G is connected. Now, by Theorem 2, $G = G_1 \oplus^1 \cdots \oplus^{\ell-1} G_\ell$ where each G_i is isomorphic to a wheel graph, *Prism*, K_2, K_3, or $K_{3,3}$, and each operation $\oplus^i$ is $\oplus_1$, $\oplus_2$ or $\oplus_2^-$. We prove the claim by induction on ℓ.

If $\ell = 1$ then G is either K_2, K_3, $K_{3,3}$, prism and a wheel graph. If G is either K_2, K_3, $K_{3,3}$ or prism, then it has constant size and hence a constant number of bonds. Thus, the augmented bond polytope $\mathrm{BOND}(G, E')$ has constant size. If G' is a wheel, then applying Lemma 10 gives us the desired result.

The inductive step. Let $G' = G_1 \oplus^1 \cdots \oplus^{\ell-2} G_{\ell-1}$. Then $G = G' \oplus^{\ell-1} G_\ell$.

If $G = G' \oplus_1 G_\ell$, then Lemma 6 with Theorem 3 gives us the desired result.

If $G = G' \oplus_e G_\ell$, then Theorem 8 gives us the desired result.

Finally, if $G = G' \oplus_e^- G_\ell$, then Lemma 9, Part 1 or Part 2 – depending on how the end vertices of e are connected in $G_1 \setminus e$ and $G_2 \setminus e$ – gives us the desired result. □

Applying Theorem 11 to $(K_5 \setminus e)$-minor-free graphs together with $E' = \emptyset$ we get the following result.

Theorem 12. *Let $G = (V, E)$ be a $(K_5 \setminus e)$-minor-free graph. Then,* $\mathrm{xc}(\mathrm{BOND}(G)) \leqslant \mathcal{O}(|V|)$. *Moreover, this extended formulation can be constructed in time $\mathcal{O}(|V|)$.*

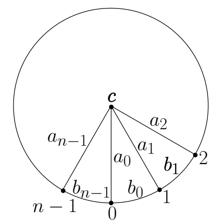
4 The Algorithm

We use the same framework as Chimani et al. [6] did, based on the fact that decomposition of G into 3-connected components can be constructed in linear time due to the algorithm of Hopcroft and Tarjan [16]. The important key difference is that for the wheel graph, we describe a relatively simple linear time algorithm whereas Chimani et al. rely on the algorithm for the construction of maximum bonds on graphs of bounded treewidth [9]. The main idea of our algorithm is simple - to mimic Kadane's dynamic programming approach [3] for the Maximum sum subarray problem in a slightly more complicated setting.

Theorem 13. *The* MAX-BOND *problem can be solved in time $O(n)$ for the wheel graph W_n.*

Proof. Given a weighted wheel graph W_n, for notational simplicity, we also use the following notation: for $i = 0, 1, \ldots, n-2$, $a_i = w(i, c)$, $b_i = w(i, i+1)$, and $a_{n-1} = w(n-1, c)$ and $b_{n-1} = w(n-1, 0)$.

Given a bond $(S, V \setminus S)$ of W_n, its two connected components have a very special form: either one of them is the hub vertex c and the other consists of all the vertices on the rim - such a bond is called the *trivial bond*, or one of the connected components is a path on the rim of length at most $n-2$, denoted P_S in the following, and the other component consists of all the other vertices which is a fan graph.



Let $\mathcal{F}$ be the set of all non-trivial bonds in W_n. Consider the partition of $\mathcal{F}$ into $\mathcal{F}_1 \cup \mathcal{F}_2 \cup \mathcal{F}_3$ defined bellow:

$$\mathcal{F}_1 = \{S \in \mathcal{F} \mid \text{the path } P_S \text{ does not contain the edges } (n-1, 0) \text{ and } (0, 1)\}$$
$$\mathcal{F}_2 = \{S \in \mathcal{F} \mid \text{the path } P_S \text{ does not contain the edges } (0, 1) \text{ and } (1, 2)\}$$
$$\mathcal{F}_3 = \{S \in \mathcal{F} \mid \text{the path } P_S \text{ contains the edge } (0, 1)\}$$

Thus, if we can find for each i, $\min_{S \in \mathcal{F}_i} \sum_{e \in \delta(S)} w(e)$, in linear time, we can solve the bond problem on the wheel graph in linear time. Also note that the sets $\mathcal{F}_1$ and $\mathcal{F}_2$ are of the same type, just *rotated*; thus, it suffices to describe an algorithm for finding the optimal bond from $\mathcal{F}_1$, and from $\mathcal{F}_3$.

Finding the optimal bond from $\mathcal{F}_1$. For each $k \in \{1, n-1\}$, we define the following quantities; for most of the quantities we introduce two names - a full name, indicating its meaning, and an abbreviation:

$$\begin{aligned}
&\mathsf{BSF}(k) = \mathsf{Best\text{-}So\text{-}Far}(k) = \max\{b_{i-1} + b_j + \sum_{l=i}^{j} a_l \ : \ i \in [1, k], j \in [i, k]\} \\
&\mathsf{BSFL}(k) = \mathsf{Best\text{-}So\text{-}Far\text{-}ind\text{-}L}(k) = \\
&\qquad\qquad \min\bigl\{i \in [1, k] \ : \ \mathsf{BSF}(k) = \max\{b_{i-1} + b_j + \sum_{l=i}^{j} a_l \ : \ j \in [i, k]\}\bigr\} \\
&\mathsf{BSFR}(k) = \mathsf{Best\text{-}So\text{-}Far\text{-}ind\text{-}R}(k) = \\
&\qquad\qquad \min\{j \in [\mathsf{BSFL}(k), k] \ : \ \mathsf{BSF}(k) = b_{\mathsf{BSFL}(k)-1} + b_j + \sum_{l=\mathsf{BSFL}(k)}^{j} a_l\} \\
&\mathsf{SB}(k) = \mathsf{Suffix\text{-}Best}(k) = \max\{b_{i-1} + b_k + \sum_{l=i}^{k} a_l \ : \ i \in [1, k]\} \\
&\mathsf{Suffix\text{-}Best\text{-}ind\text{-}L}(k) = \min\{i \in [1, k] \ : \ \mathsf{SB}(k) = b_{i-1} + b_k + \sum_{l=i}^{k} a_l\}
\end{aligned}$$

In words, $\mathsf{Best\text{-}So\text{-}Far}(k)$ is the cost of the maximum bond that cuts out a subpath of $\{1, 2, \ldots, k\}$, and the numbers $\mathsf{Best\text{-}So\text{-}Far\text{-}ind\text{-}L}(k)$ and $\mathsf{Best\text{-}So\text{-}Far\text{-}ind\text{-}R}(k)$ are the indices i and j for which the maximum value $\mathsf{Best\text{-}So\text{-}Far}(k)$ is attained; as there might be more such indices, we pick the smallest ones. Similarly, $\mathsf{Suffix\text{-}Best}(k)$ is the cost of the maximum bond that cuts out a subpath of $\{1, 2, \ldots, k\}$ ending in k, and the number $\mathsf{Suffix\text{-}Best\text{-}ind\text{-}L}(k)$ is the index of the vertex in which the subpath of the cost $\mathsf{Suffix\text{-}Best}(k)$ starts. Note that $\mathsf{Best\text{-}So\text{-}Far}(n-1)$ is the cost of the maximal bond from the set $\mathcal{F}_1$ and that it consists of edges $\{\{\mathsf{Best\text{-}So\text{-}Far\text{-}ind\text{-}L}(n-1)-1, \mathsf{Best\text{-}So\text{-}Far\text{-}ind\text{-}L}(n-1)\}, \{\mathsf{Best\text{-}So\text{-}Far\text{-}ind\text{-}R}(n-1), \mathsf{Best\text{-}So\text{-}Far\text{-}ind\text{-}R}(n-1)+1\}\} \cup \bigcup_{l=\mathsf{BSFL}(n-1)}^{\mathsf{BSFR}(n-1)} \{l, c\}$.

The quantities can be computed in linear time using dynamic programming by Procedure Best-Bond-1.

Procedure 1 Best-Bond-1

$\mathsf{Best\text{-}So\text{-}Far}(1) \leftarrow b_{n-1} + b_1 + a_1, \mathsf{Suffix\text{-}Best}(1) \leftarrow \mathsf{Best\text{-}So\text{-}Far}(1)$
$\mathsf{Best\text{-}So\text{-}Far\text{-}ind\text{-}L}(1) \leftarrow 1,\ \mathsf{Best\text{-}So\text{-}Far\text{-}ind\text{-}R}(1) \leftarrow 1,\ \mathsf{Suffix\text{-}Best\text{-}ind\text{-}L}(1) \leftarrow 1$
for $j = 1, \dots, n-2$ **do**
  $\mathsf{New\text{-}Best\text{-}Candidate} \leftarrow \mathsf{Suffix\text{-}Best}(j) + a_{j+1} - b_j + b_{j+1}$
  if $\mathsf{Best\text{-}So\text{-}Far}(j) \geq \mathsf{New\text{-}Best\text{-}Candidate}$ **then**
    $\mathsf{Best\text{-}So\text{-}Far}(j+1) \leftarrow \mathsf{Best\text{-}So\text{-}Far}(j)$
    $\mathsf{Best\text{-}So\text{-}Far\text{-}ind\text{-}L}(j+1) \leftarrow \mathsf{Best\text{-}So\text{-}Far\text{-}ind\text{-}L}(j)$
    $\mathsf{Best\text{-}So\text{-}Far\text{-}ind\text{-}R}(j+1) \leftarrow \mathsf{Best\text{-}So\text{-}Far\text{-}ind\text{-}R}(j)$
  else
    $\mathsf{Best\text{-}So\text{-}Far}(j+1) \leftarrow \mathsf{New\text{-}Best\text{-}Candidate}$
    $\mathsf{Best\text{-}So\text{-}Far\text{-}ind\text{-}L}(j+1) \leftarrow \mathsf{Suffix\text{-}Best\text{-}ind\text{-}L}(j)$
    $\mathsf{Best\text{-}So\text{-}Far\text{-}ind\text{-}R}(j+1) \leftarrow j+1$
  if $\mathsf{Suffix\text{-}Best}(j) < 0$ **then**
    $\mathsf{Suffix\text{-}Best}(j+1) \leftarrow a_{j+1} - b_j + b_{j+1}$
    $\mathsf{Suffix\text{-}Best\text{-}ind\text{-}L}(j+1) \leftarrow j+1$
  else
    $\mathsf{Suffix\text{-}Best}(j+1) \leftarrow \mathsf{Suffix\text{-}Best}(j) + a_{j+1} - b_j + b_{j+1}$
    $\mathsf{Suffix\text{-}Best\text{-}ind\text{-}L}(j+1) \leftarrow \mathsf{Suffix\text{-}Best}(j)$
return$(\mathsf{Best\text{-}So\text{-}Far}(n-1), \mathsf{Best\text{-}So\text{-}Far\text{-}ind\text{-}L}(n-1), \mathsf{Best\text{-}So\text{-}Far\text{-}ind\text{-}R}(n-1))$

Finding the optimal bond from $\mathcal{F}_3$. For each $k \in [1, n-2]$, we define the following quantities:

$$\mathsf{Prefix\text{-}Best\text{-}Right}(k) = \max\{b_j + \sum_{l=1}^{j} a_l \ : \ j \in [1,k]\}$$
$$\mathsf{Prefix\text{-}Best\text{-}Right\text{-}ind}(k) = \min\{j \in [1,k] \ : \ \mathsf{Prefix\text{-}Best\text{-}Right}(k) = b_j + \sum_{l=1}^{j} a_l\}$$
$$\mathsf{Prefix\text{-}Right}(k) = b_k + \sum_{l=1}^{k} a_l$$

and for each $k \in [3, n]$, we define and then backwards calculate the following ones:

$$\mathsf{Prefix\text{-}Best\text{-}Left}(k) = \max\{b_{k-1} + \sum_{l=j}^{n-1} a_l + a_0 \ : \ j \in [k,n]\}$$
$$\mathsf{Prefix\text{-}Best\text{-}Left\text{-}ind}(k) = \min\{j \in [k,n] \ : \ \mathsf{Prefix\text{-}Best\text{-}Left}(k) = b_{j-1} + \sum_{l=j}^{n-1} a_l + a_0\}$$

$$\mathsf{Prefix\text{-}Left}(k) = b_{k-1} + \sum_{l=k}^{n-1} a_l$$

Similarly as before, these quantities can be computed in linear time using dynamic programming (formal description in the Appendix). We observe two things: for every bond F of type 3, there exists a vertex $i \in \{2, \ldots, n-1\}$ such that i does not belong to the path that is cut off by the bond S. If S is the optimal bond of type 3 and i is the vertex not belonging to the path cut off by it, then the cost of S equals $\mathsf{Prefix\text{-}Best\text{-}Right}(i) + \mathsf{Prefix\text{-}Best\text{-}Left}(i)$. Thus, the cost of the optimal bond of type 3 is $\max\{\mathsf{Prefix\text{-}Best\text{-}Right}(i) + \mathsf{Prefix\text{-}Best\text{-}Left}(i) : i \in \{2, \ldots, n-1\}\}$.

To obtain the optimal bond, we compare the weights (costs) of the trivial bond and the optimal bonds of types 1, 2, and 3, and pick as our solution the best one. Note that the total running time is $\mathcal{O}(n)$ only. □

Combining Theorem 13 with the algorithm of Chimani et al. [6] (cf. Theorem 2 and Lemma 5), we obtain the following result.

Corollary 14. *The* Max-Bond *problem can be solved for any $(K_5 \setminus e)$-minor-free graph in time $O(n)$.*

For the sake of completeness, a self-contained proof of the Corollary 14 is also given in the Appendix.

5 Concluding Remarks

Our main result concerning k-sums for $k = 1, 2$ can be used in a natural way to get explicit descriptions of the bond polytope of the resulting graph. Let $G = G_1 \oplus_1 G_2$. Then Lemma 6 allows us to explicitly obtain the inequalities describing $\mathrm{BOND}(G)$ since it is just a subdirect sum of the two polytopes $\mathrm{BOND}(G_1)$ and $\mathrm{BOND}(G_2)$. We include a complete description of these inequalities in the Appendix (Lemma 15). Unfortunately, the number of inequalities is not additive, and this cannot be avoided unless one constructs extended formulations, as we do.

One can also construct the inequalities describing $\mathrm{BOND}(G_1 \oplus_2 G_2)$ by first constructing the extended formulation in Theorem 8 and then projecting out the extra coordinates that were added. Since there is only a constant number of extra coordinates that need to be projected out, this can be done in polynomial time. However, in general, one would have neither a linear size description nor a linear (in output size) time construction.

Finally, for the 2-sum operation where the common edge is removed, we believe that the extension complexity of $\mathrm{BOND}\left(G_1 \oplus_2^- G_2\right)$ is not additive in the extension complexities of $\mathrm{BOND}(G_1)$ and $\mathrm{BOND}(G_2)$. This is because in Lemma 9 we need the bond polytopes not of the summand graphs but of their subgraphs.

References

1. E. Balas. Disjunctive programming: Properties of the convex hull of feasible points. *Discret. Appl. Math.*, 89(1-3):3–44, 1998.
2. F. Barahona and A. R. Mahjoub. On the cut polytope. *Math. Program.*, 36(2):157–173, 1986.
3. J. L. Bentley. Programming pearls: algorithm design techniques. *Communications of The ACM*, 27:865–873, 1984.
4. R. Carvajal, M. Constantino, M. Goycoolea, J. P. Vielma, and A. Weintraub. Imposing connectivity constraints in forest planning models. *Oper. Res.*, 61(4):824–836, 2013.
5. B. Chaourar. Connected max cut is polynomial for graphs without the excluded minor $K_5 \setminus e$. *J. Comb. Optim.*, 40(4):869–875, 2020.
6. M. Chimani, M. Juhnke-Kubitzke, and A. Nover. On the bond polytope. *Advances in Geometry*, 23(4):461–480, 2023.
7. B. Courcelle. The monadic second-order logic of graphs I: Recognizable sets of finite graphs. *Information and Computation*, 85:12–75, 1990.
8. G. L. Duarte, H. Eto, T. Hanaka, Y. Kobayashi, Y. Kobayashi, D. Lokshtanov, L. L. C. Pedrosa, R. C. S. Schouery, and U. S. Souza. Computing the largest bond and the maximum connected cut of a graph. *Algorithmica*, 83(5):1421–1458, 2021.
9. G. L. Duarte, D. Lokshtanov, L. L. C. Pedrosa, R. C. S. Schouery, and U. S. Souza. Computing the Largest Bond of a Graph. In *Proc. of 14th International Symposium on Parameterized and Exact Computation (IPEC)*, volume 148 of *LIPIcs*, pages 12:1–12:15, 2019.
10. H. Eto, T. Hanaka, Y. Kobayashi, and Y. Kobayashi. Parameterized algorithms for maximum cut with connectivity constraints. In *Proc. of 14th International Symposium on Parameterized and Exact Computation (IPEC)*, volume 148 of *LIPIcs*, pages 13:1–13:15, 2019.
11. R. Gandhi, M. T. Hajiaghayi, G. Kortsarz, M. Purohit, and K. K. Sarpatwar. On maximum leaf trees and connections to connected maximum cut problems. *Inf. Process. Lett.*, 129:31–34, 2018.
12. N. Garg and V. V. Vazirani. A polyhedron with all s-t cuts as vertices, and adjacency of cuts. *Math. Program.*, 70:17–25, 1995.
13. V. Grimm, T. Kleinert, F. Liers, M. Schmidt, and G. Zöttl. Optimal price zones of electricity markets: a mixed-integer multilevel model and global solution approaches. *Optim. Methods Softw.*, 34(2):406–436, 2019.
14. M. Hajiaghayi, G. Kortsarz, R. MacDavid, M. Purohit, and K. K. Sarpatwar. Approximation algorithms for connected maximum cut and related problems. *Theor. Comput. Sci.*, 814:74–85, 2020.
15. J. Hopcroft and R. Tarjan. Algorithm 447: efficient algorithms for graph manipulation. *Commun. ACM*, 16(6):372–378, June 1973.
16. J. Hopcroft and R. Tarjan. Dividing a graph into triconnected components. *SIAM J. Comput.*, 2(3):135–158, 1973.
17. R. M. Karp. Reducibility among combinatorial problems. In *Proc. of a symposium on the Complexity of Computer Computations*, pages 85–103, 1972.
18. P. Kolman, M. Koutecký, and H. R. Tiwary. Extension Complexity, MSO Logic, and Treewidth. *Discrete Mathematics & Theoretical Computer Science*, vol. 22 no. 4, Oct. 2020.
19. A. Langer, F. Reidl, P. Rossmanith, and S. Sikdar. Practical algorithms for MSO model-checking on tree-decomposable graphs. *Computer Science Review*, 13-14:39–74, 2014.

20. F. Margot. *Composition de polytopes combinatoires: une approche par projection.* PhD thesis, École polytechnique fédérale de Lausanne, 1994.
21. P. McMullen. Constructions for projectively unique polytopes. *Discrete Mathematics*, 14(4):347–358, 1976.
22. B. Schieber and S. Vahidi. Approximating connected maximum cuts via local search. In *Proc. of 31st Annual European Symposium on Algorithms (ESA)*, volume 274 of *LIPIcs*, pages 93:1–93:17, 2023.
23. M. Skutella and A. Weber. On the dominant of the s-t-cut polytope: Vertices, facets, and adjacency. *Math. Program.*, 124(1-2):441–454, 2010.
24. S. Vicente, V. Kolmogorov, and C. Rother. Graph cut based image segmentation with connectivity priors. In *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.
25. K. Wagner. Bemerkungen zu Hadwigers Vermutung. *Mathematische Annalen*, 141:433–452, 1960.

Appendix

Max-Bond on the wheel grpahs

Here we describe the second procedure used for solving the Max-Bond problem on the wheel graphs W_n in the proof of Theorem 13.

Procedure 2 Best-Bond-3

```
Prefix-Best-Right(1) ← b_1 + a_1, Prefix-Right(1) ← Prefix-Best-Right(1)
Prefix-Best-Right-ind(1) ← 1
for j = 1, ..., n − 3 do
    New-Best-Candidate ← Prefix-Right(j) + a_{j+1} − b_j + b_{j+1}
    if Prefix-Best-Right(j) ≥ New-Best-Candidate then
        Prefix-Best-Right(j + 1) ← Prefix-Best-Right(j)
        Prefix-Best-Right-ind(j + 1) ← Prefix-Best-Right-ind(j)
    else
        Prefix-Best-Right(j + 1) ← New-Best-Candidate
        Prefix-Best-Right-ind(j + 1) ← j + 1
    Prefix-Right(j + 1) ← Prefix-Right(j) + a_{j+1} − b_j + b_{j+1}
Prefix-Best-Left(n) ← b_{n−1} + a_0, Prefix-Left(n) ← Prefix-Best-Left(n)
Prefix-Best-Left-ind(n) ← n
for j = n, ..., 3 do
    New-Best-Candidate ← Prefix-Left(j) + a_{j−1} − b_{j−1} + b_{j−2}
    if Prefix-Best-Left(j) ≥ New-Best-Candidate then
        Prefix-Best-Left(j − 1) ← Prefix-Best-Left(j)
        Prefix-Best-Left-ind(j − 1) ← Prefix-Best-Left-ind(j)
    else
        Prefix-Best-Left(j − 1) ← New-Best-Candidate
        Prefix-Best-Left-ind(j − 1) ← j − 1
    Prefix-Left(j − 1) ← Prefix-Left(j) + a_{j−1} − b_{j−1} + b_{j−2}
Best-Solution ← Prefix-Best-Right(2) + Prefix-Best-Left(2)
for j = 3, ..., n − 1 do
    New-Best-Candidate ← Prefix-Best-Right(j) + Prefix-Best-Left(j)
    if Best-Solution < New-Best-Candidate then
        Best-Solution ← New-Best-Candidate
return(Best-Solution)
```

Proof of Theorem 2

Proof of Theorem 2. We start by finding a decomposition of the $(K_5 \setminus e)$-minor-free graph G into a tree of 2-connected components; this can be done in linear time by a depth-first search algorithm [15]. The components are attached to each other at shared vertices and G corresponds to 1-sums of these components.

Consider now a 2-connected component H of G. By a linear time algorithm of Hopcroft and Tarjan [16] we construct a decomposition of H into a tree T of

3-connected components. Informally, the nodes of T are 3-connected subgraphs of H and if two nodes share an edge in T then the corresponding subgraphs in H share two vertices. By induction on the number of vertices in T we show that H is obtained from a wheel graph, Prism, K_2, K_3, and $K_{3,3}$ by the operations $\oplus_2$ and $\oplus_2^-$.

If T has only a single vertex, then H is a 3-connected $(K_5 \setminus e)$-minor-free graph; the only such graphs are a wheel graph, Prism, K_2, K_3, or $K_{3,3}$ (cf. [6]) which completes the proof of the base case.

For the inductive step, assume that T has at least two vertices, and let t be an arbitrary leaf of T. Let $H_1 = (V_1, E_1)$ be the subgraph of H corresponding to $T \setminus t$, $H_2 = (V_2, E_2)$ be the subgraph of H corresponding to t, and let u and v be the two vertices in $V_1 \cap V_2$. We distinguish two cases.

If $uv \in E_1$, then $H = H_1 \oplus_{uv} H_2$. Therefore, by our inductive hypothesis, $H_1 = G_1 \oplus^1 \cdots \oplus^{\ell-2} G_{\ell-1}$ where for each i, G_i is a wheel graph, Prism, K_2, K_3, or $K_{3,3}$, and $\oplus^i$ is either $\oplus_1, \oplus_2$, or $\oplus_2^-$. Since $G = H_1 \oplus_2 H_2$, and H_2 is one of the graphs in our list, the proof is completed.

If $uv \notin E$, then $H = H_1' \oplus_{uv}^- H_2'$ where $H_i' = H_i \cup \{uv\}$ for $i = 1, 2$. Observe that both H_1' and H_2' are $(K_5 \setminus e)$-minor-free graphs. To see this, assume without loss of generality that H_1' contains a $K_5 \setminus e$ minor. As H_2' is a connected graph, u and v are connected by a path in H_2' and so H contains a $K_5 \setminus e$ minor as well, which is a contradiction to the fact that G is $(K_5 \setminus e)$-minor-free.

Therefore, by our inductive hypothesis, $H_1' = G_1 \oplus^1 \cdots \oplus^{\ell-2} G_{\ell-1}$ where for each i, G_i is a wheel graph, Prism, K_2, K_3, or $K_{3,3}$, and $\oplus^i$ is either $\oplus_1, \oplus_2$, or $\oplus_2^-$. Since H_2' is a 3-connected $(K_5 \setminus e)$-minor-free graph, it is either a wheel graph, Prism, K_2, K_3, or $K_{3,3}$. Finally, as $G = H_1' \oplus_2^- H_2'$, the proof is completed. □

Lemma 15. *Let* $P_1 = \operatorname{conv}(\mathbf{V}_1) = \{\mathbf{x} \mid \mathbf{A}_1\mathbf{x} \leqslant \mathbf{0}, \mathbf{B}\mathbf{x} \leqslant \mathbf{1}\}$ *and* $P_2 = \operatorname{conv}(\mathbf{V}_2) = \{\mathbf{y} \mid \mathbf{A}_2\mathbf{y} \leqslant \mathbf{0}, \mathbf{C}\mathbf{y} \leqslant \mathbf{1}\}$ *be polytopes. Then,*

$$\operatorname{conv}\begin{pmatrix}\mathbf{V}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_2\end{pmatrix} = \left\{ \begin{pmatrix}\mathbf{x} \\ \mathbf{y}\end{pmatrix} \;\middle|\; \begin{array}{ll} \mathbf{A}_1\mathbf{x} \leqslant \mathbf{0}, \mathbf{A}_2\mathbf{y} \leqslant \mathbf{0} & \\ \mathbf{b}^\intercal\mathbf{x} + \mathbf{c}^\intercal\mathbf{y} \leqslant 1 & \forall \mathbf{b} \in \mathbf{B}, \forall \mathbf{c} \in \mathbf{C} \end{array} \right\}. \tag{1}$$

Proof. Let P and P', resp., denote the polytopes on the left and right, resp., sides of the equality (1). We start by showing that $P \subseteq P'$. Consider $(\mathbf{x}, \mathbf{y})^\intercal \in P$. Then $\mathbf{x} = \sum_{\mathbf{u} \in \mathbf{V}_1} \lambda_\mathbf{u} \mathbf{u}$, $\mathbf{y} = \sum_{\mathbf{v} \in \mathbf{V}_2} \lambda_\mathbf{v} \mathbf{v}$ for some nonnegative coefficients $\lambda_\mathbf{z}$, $\mathbf{z} \in \mathbf{V}_1 \cup \mathbf{V}_2$, such that $\sum_{\mathbf{u} \in \mathbf{V}_1} \lambda_\mathbf{u} + \sum_{\mathbf{v} \in \mathbf{V}_2} \lambda_\mathbf{v} = 1$. As for each $\mathbf{u} \in \mathbf{V}_1$ and $\mathbf{b} \in \mathbf{B}$ we have $\mathbf{A}_1\mathbf{u} \leqslant \mathbf{0}$ and $\mathbf{b}^\intercal\mathbf{u} \leqslant 1$, and analogously, for each $\mathbf{v} \in \mathbf{V}_2$ and $\mathbf{c} \in \mathbf{C}$ we have $\mathbf{A}_2\mathbf{v} \leqslant \mathbf{0}$ and $\mathbf{c}^\intercal\mathbf{v} \leqslant 1$, it holds $\mathbf{A}_1\mathbf{x} \leqslant \mathbf{0}$, $\mathbf{A}_2\mathbf{y} \leqslant \mathbf{0}$ and $\mathbf{b}^\intercal\mathbf{x} + \mathbf{c}^\intercal\mathbf{y} \leqslant 1$. Thus, $P \subseteq P'$.

Consider now $(\mathbf{x}, \mathbf{y})^\intercal \in P'$. If for some $\mathbf{b} \in \mathbf{B}$, $\mathbf{b}^\intercal\mathbf{x} \geq 0$, then for every $\mathbf{c} \in \mathbf{C}$, $\mathbf{c}^\intercal\mathbf{y} \leq 1$, that is, $\mathbf{C}\mathbf{y} \leqslant \mathbf{1}$; similarly, if for some $\mathbf{c} \in \mathbf{C}$, $\mathbf{c}^\intercal\mathbf{y} \geq 0$, then for every $\mathbf{b} \in \mathbf{B}$, $\mathbf{b}^\intercal\mathbf{y} \leq 1$, that is, $\mathbf{B}\mathbf{x} \leqslant \mathbf{1}$. Thus, to prove that $P' \subseteq P$, it suffice to show that for some $\mathbf{b} \in \mathbf{B}$, $\mathbf{b}^\intercal\mathbf{x} \geq 0$ and for some $\mathbf{c} \in \mathbf{C}$, $\mathbf{c}^\intercal\mathbf{y} \geq 0$; note that the inequalities $\mathbf{A}_1\mathbf{x} \leqslant \mathbf{0}$ and $\mathbf{A}_2\mathbf{y} \leqslant \mathbf{0}$ are always satisfied for our $\mathbf{x}$ and $\mathbf{y}$.

Assume, for a contradiction, that for every $\mathbf{b} \in \mathbf{B}$, $\mathbf{b}^\intercal\mathbf{x} < 0$. Then not only $\mathbf{x} \in P_1$, but for every non-negative λ, also $\lambda\mathbf{x} \in P_1$. However, this is a contradiction with the fact that P_1 a polytope. Thus, there exists $\mathbf{b} \in \mathbf{B}$ such that $\mathbf{b}^\intercal \mathbf{x} \geq 0$. By the same arguments, there exists $\mathbf{c} \in \mathbf{C}$ such that $\mathbf{c}^\intercal \mathbf{y} \geq 0$. This completes the proof of the lemma. $\square$

(The above operation is called a *subdirect sum* and the inequalities follow from Proposition 2.3 in [21].)

The linear time algorithm

For the sake of completeness we provide a complete description of the linear time algorithm, building on the presentation of Chimani et al. [6]. Let $\mathrm{MAXB}(G)$ denote the size of the maximum bond in G, and given two vertices u, v from G, let $\mathrm{MAXB}^{uv}(G)$ ($\mathrm{MAXB}^{u}_{v}(G)$, resp.) denote the size of the maximum bond of G in which the vertices u, v are on the same side (on the opposite sides, resp.) of the bond.

We start by observing that given an algorithm for $\mathrm{MAXB}(G)$ running in time $p(|G|)$, for every edge $uv \in E(G)$ we can construct $\mathrm{MAXB}^{uv}(G)$ in time $p(|G|)$, and the same holds for $\mathrm{MAXB}^{u}_{v}(G)$; in the first case we let the algorithm construct $\mathrm{MAXB}(G')$ where G' is the graph obtained from G by changing the weight of the edge uv to $w(uv) = \sum_{e \in E} w(e)$, and in the second case to $w(uv) = -\sum_{e \in E} w(e)$.

We proceed with a technical lemma.

Lemma 16. *Let G_1, G_2 be 2-connected graphs, $uv \in E(G_1) \cap E(G_2)$ be an edge that appears in both of them. Then*

$$\mathrm{MAXB}(G_1 \oplus^{-}_{uv} G_2) = \max\{\mathrm{MAXB}^{uv}(G_1 \setminus uv), \mathrm{MAXB}(G'_2)\}$$
$$\mathrm{MAXB}(G_1 \oplus_{uv} G_2) = \max\{\mathrm{MAXB}^{uv}(G_1), \mathrm{MAXB}(\bar{G}_2)\}$$

where G'_2 and $\bar{G}_2$, resp., is the graph obtained from G_2 by changing the weight of the edge uv to $w(uv) = \mathrm{MAXB}^{u}_{v}(G_1 \setminus uv)$ and to $w(uv) = \mathrm{MAXB}^{u}_{v}(G_1)$, resp.

Proof. Let $G = G_1 \oplus^{-}_{uv} G_2$ and let F be a maximum bond in G. By Lemma 5, Case 2,

a) F is a bond of $G_1 \setminus uv$ with u and v on the same side, or
b) F is a bond of $G_2 \setminus uv$ with u and v on the same side, or
c) $F \cap E(G_i)$ is a bond of $G_i \setminus uv$ with u and v on different sides, for both $i \in \{1, 2\}$.

In case a), $\mathrm{MAXB}(G) = \mathrm{MAXB}^{uv}(G_1 \setminus uv) \geq \mathrm{MAXB}(G'_2)$. In case b), $\mathrm{MAXB}(G) = \mathrm{MAXB}^{uv}(G_2 \setminus uv) = \mathrm{MAXB}(G'_2) \geq \mathrm{MAXB}^{uv}(G_1 \setminus uv)$. In case c), $\mathrm{MAXB}(G) = \mathrm{MAXB}^{u}_{v}(G_1 \setminus uv) + \mathrm{MAXB}^{u}_{v}(G_2 \setminus uv) = \mathrm{MAXB}(G'_2) \geq \mathrm{MAXB}^{uv}(G_1 \setminus uv)$. Thus, in all three cases, we have the desired equality

$$\mathrm{MAXB}(G_1 \oplus^{-}_{uv} G_2) = \max\{\mathrm{MAXB}^{uv}(G_1 \setminus uv), \mathrm{MAXB}(G'_2)\} .$$

If $G = G_1 \oplus_{uv} G_2$, we proceed in a similar way, using the fact (cf. [5,6]) that F is a bond in G if and only if

a) F is a bond of G_1 with u and v on the same side, or
b) F is a bond of G_2 with u and v on the same side, or
c) $F \cap E(G_i)$ is a bond of G_i with u and v on different sides, for both $i \in \{1, 2\}$.

□

Proof of Corollary 14. First we prove the claim for 2-connected graphs. For a 2-connected $(K_5 \setminus e)$-minor-free graph $G = (V, E)$, by Theorem 2, we construct in linear time its decomposition $G = G_1 \oplus^1 \cdots \oplus^{l-1} G_\ell$ where each G_i is isomorphic to a wheel graph, *Prism*, K_3, or $K_{3,3}$, and each operation $\oplus^i$ is $\oplus_2$ or $\oplus_2^-$.

By induction on l we show the following: there exists a constant $c > 0$ such that given a decomposition of G into $G = G_1 \oplus^1 \cdots \oplus^{l-1} G_\ell$ where each G_i is isomorphic to a wheel graph, *Prism*, K_3, or $K_{3,3}$, and each operation $\oplus^i$ is $\oplus_2$ or $\oplus_2^-$, it is possible to compute $\mathrm{MAXB}(G)$ in time at most $2 \cdot c \cdot \sum_{i=1}^{l} |V(G_i)|$. Since, $\sum_{i=1}^{l} |V(G_i)| = |V(G)| + 2(\ell - 1)$ and $\ell \leqslant |V(G)|$ we have $\sum_{i=1}^{l} |V(G_i)| \leqslant 3 \cdot |V(G)| - 2$ and hence the upper bound on the running time will follow.

If $l = 1$, then G is a wheel graph W_n, Prism, K_3 or $K_{3,3}$; as each of them, except for W_n, is a constant size graph, and for the wheel graph W_n, $\mathrm{MAXB}(W_n)$ can be computed in linear time by Theorem 13, we conclude, considering our initial observation of this section, that there exists a constant $c > 0$ such that for any G of the graphs listed in the previous sentence and any $uv \in E(G)$, both $\mathrm{MAXB}^{uv}(G)$ and $\mathrm{MAXB}_v^u(G)$ can be computed in time at most $c \cdot |V(G)|$.

If $\ell \geq 2$, let $H = G_2 \oplus^2 \cdots \oplus^{l-1} G_\ell$. We distinguish two cases: $\oplus^1 = \oplus_{uv}^-$ and $\oplus^1 = \oplus_{uv}$. In the first case, let H' be the graph obtained from H by changing the weight of the edge uv to $w(uv) = \mathrm{MAXB}_v^u(G_1 \setminus uv)$; note that H' has the same decomposition as H, they differ only in the weight of the edge uv. Thus, by the inductive assumption, we can compute $\mathrm{MAXB}(H')$ in time $2 \cdot c \cdot \sum_{i=2}^{l} |V(G_i)|$, and $\mathrm{MAXB}^{uv}(G_1)$ and $\mathrm{MAXB}_v^u(G_1)$ in time $c \cdot |V(G_1)|$. By Lemma 16,

$$\mathrm{MAXB}(G) = \mathrm{MAXB}(G_1 \oplus_{uv}^- H) = \max\{\mathrm{MAXB}^{uv}(G_1 \setminus uv), \mathrm{MAXB}(H')\} ,$$

therefore we can compute $\mathrm{MAXB}(G)$ from $\mathrm{MAXB}^{uv}(G_1 \setminus uv)$ and $\mathrm{MAXB}(H')$ in time $\mathcal{O}(1)$. Note that the time to construct H' given H, uv and $\mathrm{MAXB}_v^u(G_1)$, is $\mathcal{O}(1)$. Thus, exploiting the inductive assumption, we can compute $\mathrm{MAXB}(G)$ in time $c \cdot |V(G_1)| + 2 \cdot c \cdot \sum_{i=2}^{l} |V(G_i)| + \mathcal{O}(1) \leq 2 \cdot c \cdot \sum_{i=1}^{l} |V(G_i)|$ which completes the proof of the inductive step in the first case.

If $\oplus^1 = \oplus_{uv}$, we proceed analogously, exploiting the other equality of Lemma 16.

Finally, if the graph is not 2-connected, we compute in linear time a decomposition of G into 2-connected components [16], construct the maximum bond for each of them in linear time, and output the largest of them; the total running time will be $\mathcal{O}(|V(G)|) + \sum_{H \in \mathcal{C}} \mathcal{O}(|V(H)|) = \mathcal{O}(|V(G)|)$ where $\mathcal{C}$ is the set of 2-connected components of G. □