Generalised 2-Circulant Inequalities for the Max-Cut Problem

Konstantinos Kaparis[a,1,*], Adam N. Letchford[b], Ioannis Mourtos[c]

[a] QMeDA Lab, Department of Business Administration, University of Macedonia, Thessaloniki, Greece.
[b] Department of Management Science, Lancaster University, Lancaster, UK.
[c] ELTRUN Research Lab, Department of Management Science & Technology, Athens University of Economics and Business, Athens, Greece.

Abstract

The *max-cut* problem is a fundamental combinatorial optimisation problem, with many applications. Poljak and Turzik found some facet-defining inequalities for the associated polytope, which we call *2-circulant* inequalities. We present a more general family of facet-defining inequalities, an exact separation algorithm that runs in polynomial time, and some computational results.

Keywords: max-cut problem, polyhedral combinatorics, cutting planes

1. Introduction

In the *max-cut problem* (MCP), we are given a (simple, loopless) undirected graph, along with a (rational) weight for each edge. The task is to partition the vertex set into two subsets, called "shores", in a way that maximises the sum of the weights of the edges that cross from one shore to the other.

The MCP is a much-studied problem in combinatorial optimisation, with a wide range of applications (see, e.g., [3, 11, 16]). Unfortunately, it is also strongly $\mathcal{NP}$-hard [13]. At present, even the best exact algorithms can solve only instances with up to 120 nodes or so (e.g., [5, 21, 23]).

The convex hull of feasible solutions to the MCP is called the *cut polytope* [4]. This polytope has been studied in depth, and many families of strong valid linear inequalities are known for it (e.g., [4, 6, 10, 17, 18, 22]). For some of those families, efficient separation algorithms are known (e.g., [4, 5, 14, 15, 17, 19, 20]).

A *2-circulant* is a graph with vertex set $\{1, \ldots, p\}$, where $p \geq 5$, and edges $\{i, i+1\}$ and $\{i, i+2\}$ for $i = 1, \ldots, p$, where indices are taken modulo p. See Figure 1 for an example. Poljak & Turzik [22] showed that any 2-circulant with $p \equiv 1 \bmod 4$ yields a facet-defining inequality for the cut polytope. We will follow [15] in calling these inequalities *2-circulant* (2C) inequalities. Separation algorithms for them are given in [15, 19, 20].

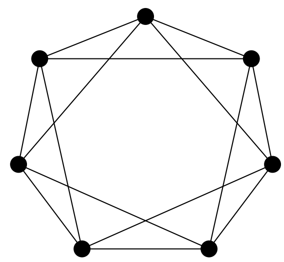

Figure 1: A 2-circulant with $p = 7$.

In this paper, we derive a more general family of facet-defining inequalities, which we call *generalised 2-circulant* (G2C) inequalities. We then show how to extend the separation algorithm in [15], in order to separate exactly over a family of valid inequalities that includes all G2C inequalities. We also present some computational results.

The paper has a simple structure. The literature is reviewed in Section 2. The new inequalities are presented in Section 3. The separation algorithm is described in Section 4, and the computational results are in Section 5.

Throughout the paper, we assume that the reader is familiar with the polyhedral approach to combinatorial

*Corresponding author
Email addresses: k.kaparis@uom.edu.gr (Konstantinos Kaparis), A.N.Letchford@lancaster.ac.uk (Adam N. Letchford), mourtos@aueb.gr (Ioannis Mourtos)

[1] This work is part of a project that has received funding from the Research Committee of the University of Macedonia under the Basic Research 2020-21 funding programme

optimisation (see, e.g., [8]). We also use the following notation and terminology. We let $K_n = (V_n, E_n)$ denote the complete graph on n nodes, where $V_n = \{1, \ldots, n\}$. Given a set $S \subseteq V$, the set of edges having exactly one end-node in S is called a *cut* and denoted by $\delta(S)$. A set $C \subset E_n$ is called a *cycle* if it induces a connected subgraph in which all nodes have even degree. A cycle is called a *circuit* if all nodes in the subgraph have degree 2. Given two sets $S, T \subseteq V_n$, the set $(S \cup T) \setminus (S \cap T)$ is called the *symmetric difference* of S and T, and is denoted by $S \triangle T$. Given a vector $x \in [0,1]^{E_n}$ and an edge $e = \{i, j\} \in E_n$, we sometimes write x_{ij} or $x(i, j)$ instead of x_e. Finally, we assume $n \geq 5$ throughout the paper, to avoid trivial or "degenerate" cases.

2. Literature Review

We now briefly review the relevant literature. We refer the reader to [11, 16] for comprehensive surveys on the MCP.

2.1. The cut polytope

For a given n and a given edge-weight vector $w \in \mathbb{Q}^{E_n}$, the MCP can be formulated as follows:

$$\begin{array}{lllr} \max & \sum_{1 \leq i < j \leq n} w_{ij} x_{ij} & & \\ \text{s.t.} & x_{ij} + x_{ik} + x_{jk} \leq 2 & (1 \leq i < j < k \leq n) & (1) \\ & x_{ij} - x_{ik} - x_{jk} \leq 0 & (\{i,j\} \in E_n;\ k \in V_n \setminus \{i,j\}) & (2) \\ & x_e \in \{0,1\} & (e \in E_n). & \end{array}$$

Here, x_e takes the value 1 if and only if edge e lies in the cut. The inequalities (1) and (2) are called *triangle* inequalities.

The convex hull of feasible x vectors is called the *cut polytope* and is denoted by CUT_n. In [4], the triangle inequalities are shown to define facets of CUT_n, along with several other inequalities, such as the *odd bicycle wheel* (OBW) inequalities. Since then, many more families have been discovered (see Part V of [11]). Here, we focus on the *2-circulant* inequalities [22], which take the form

$$\sum_{e \in F} x_e \ \leq\ 3(p-1)/2,$$

where the edge set $F \subset E_n$ induces a 2-circulant in K_n, with $p \equiv 1 \bmod 4$.

We will also need the following result from [4]: given any circuit $C \subset E_n$, and any $D \subseteq C$ with $|D|$ odd, the *co-circuit* inequality

$$\sum_{e \in D} x_e - \sum_{e \in C \setminus D} x_e \leq |D| - 1$$

is implied by the triangle inequalities.

2.2. Switching and collapsing

It is shown in [4] that, if the inequality $\lambda^T x \leq \gamma$ is valid (or facet-defining) for CUT_n, then the 'switched' inequality

$$\sum_{e \in E_n \setminus \delta(S)} \lambda_e x_e - \sum_{e \in \delta(S)} \lambda_e x_e \leq \gamma - \sum_{e \in \delta(S)} \lambda_e$$

is also valid (or facet-defining), for any $S \subset V_n$. This operation is called *switching* [11]. One can check the following facts: (i) switching on S is equivalent to switching on $V \setminus S$, (ii) switching on S and then switching on T is equivalent to switching on $S \triangle T$, (iii) if we take a triangle inequality of the form (1) and switch on node k, then we obtain a triangle inequality of the form (2).

We will also need the following fact from [9]. Let $\alpha^T x \leq \beta$ be valid for CUT_n, and let $\{i, j\}$ be an edge in E_n. We can obtain a valid inequality for CUT_{n-1} as follows. The edge $\{i, j\}$ is contracted, by identifying j with i. For any $k \in \{1, \ldots, n\} \setminus \{i, j\}$, the coefficient of x_{ik} in the new inequality is set to $\alpha_{ik} + \alpha_{jk}$. The coefficients for the edges that were not incident on i and j remain unchanged. This operation is called *collapsing*.

2.3. Separation

Separation algorithms for the cut polytope can be found in, e.g., [5, 11, 14, 15, 19, 20]. Poljak & Turzik [22] conjectured that separation for the 2C inequalities is $\mathcal{NP}$-hard. As far as we know, this conjecture remains open. On the other hand, polynomial-time separation algorithms are known for various families of valid inequalities that *include* the 2C inequalities.

Letchford [19] showed that every switched OBW or 2C inequality is implied by triangle inequalities and a *simple disjunction* of the form $(x_e = 0) \vee (x_e = 1)$. Let us call inequalities that can be derived in this way *simple disjunctive cuts* (SDCs). Using results in [2], one can separate over all SDCs by solving $\binom{n}{2}$ linear programs (LPs), each with $O(n^3)$ variables and $O(n^2)$ constraints.

A faster separation algorithm was provided by Letchford & Sørensen [20]. They defined a family of "$\{0, \frac{1}{2}\}$-cuts" for the cut polytope (see [7]), and showed that it includes the switched OBW and 2C inequalities. They then showed how to separate over the $\{0, \frac{1}{2}\}$-cuts in $O(n^5)$ time.

An alternative $O(n^5)$ separation algorithm was given in Kaparis & Letchford [15]. It separates over not only the switched 2C inequalities, but all inequalities that can be obtained from them via the collapsing operation.

3. More Facets from 2-Circulants

In this section, we derive and analyse the G2C inequalities. In Subsection 3.1, the inequalities are derived and the effect of switching is analysed. In Subsection 3.2, the G2C inequalities are shown to be intermediate in generality between the switched 2C inequalities and the SDCs. Then, in Subsection 3.3, the G2C inequalities are shown to define facets of CUT_n.

3.1. Derivation and the effect of switching

Before presenting our new inequalities, we will need the following two lemmas.

Lemma 1. *Given any ordered triple (i, j, k) of distinct vertices in V_n, the "weakened triangle" inequalities*

$$x_{ij} + x_{jk} + 2x_{ik} \le 3 \quad (3)$$

$$x_{ij} - x_{jk} - 2x_{ik} \le 0 \quad (4)$$

$$-x_{ij} + x_{jk} - 2x_{ik} \le 0 \quad (5)$$

$$-x_{ij} - x_{jk} + 2x_{ik} \le 1 \quad (6)$$

are valid for CUT_n.

Proof. The first inequality is the sum of the triangle inequality (1) and the trivial upper bound $x_{ik} \le 1$. The other three inequalities can be obtained from the first by switching on k, i and j, respectively. □

Lemma 2. *Let $C \subset E_n$ be a circuit, and let D be an arbitrary subset of C. If x is the incidence vector of a cut, then the quantity*

$$\sum_{e \in C \setminus D} x_e - \sum_{e \in D} x_e$$

is an even integer.

Proof. This follows trivially from the fact that every cut intersects every circuit an even number of times. □

The following theorem introduces the new inequalities.

Theorem 1. *Let p be an odd integer with $5 \le p \le n$. Let $v_1, \dots, v_p$ be distinct vertices in V_n. Define the set $S = \{1, \dots, p\}$, and let S^+ be an arbitrary (possibly empty) subset of S. Let S^- denote $S \setminus S^+$, and define the sets*

$$S^{++} = \{i \in S^+ : i + 1 \in S^+\},$$
$$S^{+-} = \{i \in S^+ : i + 1 \in S^-\},$$
$$S^{-+} = \{i \in S^- : i + 1 \in S^+\},$$
$$S^{--} = \{i \in S^- : i + 1 \in S^-\},$$

where indices are taken modulo p. If $3|S^{++}| + |S^{--}| \equiv 3 \bmod 4$, then the "generalised 2-circulant" (G2C) inequality

$$\sum_{i \in S^+} x(v_i, v_{i+1}) - \sum_{i \in S^-} x(v_i, v_{i+1}) + \sum_{i \in S^{++} \cup S^{--}} x(v_i, v_{i+2}) - \sum_{i \in S^{+-} \cup S^{-+}} x(v_i, v_{i+2}) \le \left(3|S^{++}| + |S^{--}| - 3\right)/2 \quad (7)$$

is valid for CUT_n (indices taken modulo p).

Proof. We assume w.l.o.g. that $v_i = i$ for $i = 1, \dots, p$, and we let L denote the left-hand side of (7). By Lemma 1, the following inequalities are valid:

$$x(i, i+1) + x(i+1, i+2) + 2x(i, i+2) \le 3 \qquad (i \in S^{++})$$
$$x(i, i+1) - x(i+1, i+2) - 2x(i, i+2) \le 0 \qquad (i \in S^{+-})$$
$$-x(i, i+1) + x(i+1, i+2) - 2x(i, i+2) \le 0 \qquad (i \in S^{-+})$$
$$-x(i, i+1) - x(i+1, i+2) + 2x(i, i+2) \le 1 \qquad (i \in S^{--}).$$

Summing these inequalities, we obtain:

$$2L \le 3|S^{++}| + |S^{--}|.$$

Dividing by two and rounding down the right-hand side, we obtain:

$$L \le \left\lfloor \frac{3|S^{++}| + |S^{--}|}{2} \right\rfloor. \quad (8)$$

Now, observe that L can be written as the sum of two components:

(*a*) $\displaystyle\sum_{i \in S^+} x(i, i+1) - \sum_{i \in S^-} x(i, i+1)$;

(*b*) $\displaystyle\sum_{i \in S^{++} \cup S^{--}} x(i, i+2) - \sum_{i \in S^{+-} \cup S^{-+}} x(i, i+2)$.

By Lemma 2, each of these two components must be an even integer. Thus, if the right-hand side of (8) is odd, we can subtract one while maintaining validity. □

Figure 2 gives a graphical representation of a G2C inequality with $p = 7$, $S = \{1, \dots, 7\}$ and $S^+ = \{1, 2, 3, 5\}$. Solid and dotted lines indicate edges whose variables have a coefficient of 1 and −1, respectively. Note that, for this example, $S^- = \{4, 6, 7\}$, $S^{++} = \{1, 2\}$ and $S^{--} = \{6\}$. Thus, $3|S^{++}| + |S^{--}| = 6 + 1 = 7$. Thus, the right-hand side is $(7 - 3)/2 = 2$.

The following proposition will turn out to be useful in the following two subsections.

Proposition 1. *Consider a fixed G2C inequality of the form (7), and let k be any element of S. Switching the G2C inequality on $\{v_k\}$ is equivalent to changing S^+ to $S^+ \triangle \{k, k-1\}$ (and adjusting S^-, S^{++} and so on accordingly).*

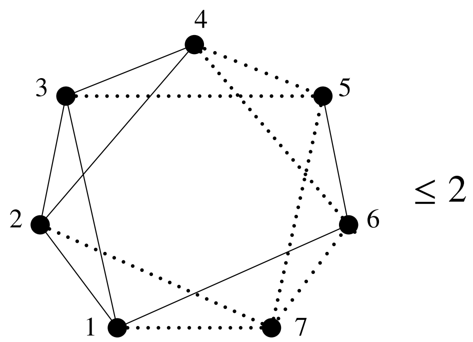

Figure 2: G2C inequality with $S = \{1,\ldots,7\}$ and $S^+ = \{1,2,3,5\}$.

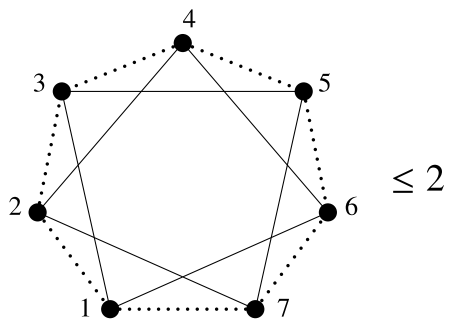

Figure 3: "Simple" G2C inequality with $p = 7$.

Proof. Suppose initially that $\{v_{k-2},\ldots,v_{k+2}\} \subseteq S^+$. Consider the modified G2C inequality that is obtained by removing v_k and v_{k-1} from S^+. The coefficients of $x(v_{k-2},v_k)$, $x(v_{k-1},v_k)$, $x(v_k,v_{k+1})$ and $x(v_k,v_{k+2})$ change from 1 to −1, and the right-hand side decreases by 4. Thus, the modified G2C inequality can be obtained from the original by switching on $\{v_k\}$. A similar argument applies if some of $v_{k-2},\ldots,v_{k+2}$ do not lie in S^+, as one can verify by an easy (but tedious) enumeration of cases. □

3.2. Relationship with other inequalities

We now compare the G2C inequalities with some other known inequalities. Let us say that an inequality is a *switched 2-circulant* (S2C) inequality if it is either a 2C inequality, or can be obtained from one by switching. The next lemma will be used to show that the S2C inequalities are a special case of the G2C inequalities.

Lemma 3. *A G2C inequality has* $p \equiv 1 \bmod 4$ *if and only if* $|S^+|$ *is odd and* $|S^-|$ *is even.*

Proof. Theorem 1 states that any G2C inequality must satisfy $3|S^{++}| + |S^{--}| \equiv 3 \bmod 4$. Moreover, by definition, we have $|S^{+-}| = |S^{-+}|$, which implies $3|S^{+-}| + |S^{-+}| \equiv 0 \bmod 4$. Put together, these imply

$$3|S^{++} \cup S^{+-}| + |S^{-+} \cup S^{--}| \equiv 3 \bmod 4,$$

or, equivalently, $3|S^+|+|S^-| \equiv 3 \bmod 4$. Now, $p = |S^+|+|S^-|$, which implies that $p + 2|S^+| \equiv 3 \bmod 4$. Given that p is odd, there are only two possibilities: either $p \equiv 1 \bmod 4$, $|S^+|$ is odd and $|S^-|$ is even, or $p \equiv 3 \bmod 4$, $|S^+|$ is even and $|S^-|$ is odd. □

Proposition 2. *A valid inequality for* CUT_n *is an S2C inequality if and only if it is a G2C inequality with* $p \equiv 1 \bmod 4$.

Proof. One can check that a 2C inequality is nothing but a G2C inequality with $p \equiv 1 \bmod 4$ and $S^+ = S$. (Indeed, in this case, we have $S^{++} = S$ and $S^{+-} = S^{-+} = S^{--} = \emptyset$, and the right-hand side of (7) reduces to $3(p-1)/2$.) Now, switching a 2C inequality on some set $T \subseteq \{v_1,\ldots,v_p\}$ is equivalent to switching on each node in T consecutively, in any order. Proposition 1 then implies that any S2C inequality is a G2C inequality.

To complete the proof, consider a G2C inequality with $p \equiv 1 \bmod 4$. Suppose $|S^-| \geq 2$, and recall from Lemma 3 that $|S^-|$ is even. Let v_k, v_ℓ be two elements of S^-, with $k < \ell$. If we switch the inequality on the set $\{v_{k+1},\ldots,v_\ell\}$, the effect is that v_k and v_ℓ move from S^- to S^+. This operation can be repeated until $S^+ = S$, at which point we have a 2C inequality. □

Before presenting our next result, we will need the following definition:

Definition 1. *A G2C inequality will be called "simple" if* $S = \{1,\ldots,p\}$, $p \equiv 3 \bmod 4$ *and* $S^+ = \emptyset$.

One can check that a simple G2C inequality takes the form:

$$-\sum_{i=1}^{p} x(i,i+1) + \sum_{i=1}^{p} x(i,i+2) \leq (p-3)/2. \quad (9)$$

See Figure 3 for a representation of a simple G2C inequality with $p = 7$. One can check that every G2C inequality with $p \equiv 3 \bmod 4$ is either simple, or can be obtained from a simple inequality by switching.

We can now present our next result.

Proposition 3. *Every G2C inequality is an SDC.*

Proof. Since the result was already shown for switched 2C inequalities in [19], it suffices to show it for the G2C inequalities with $p \equiv 3 \bmod 4$. By relabelling the nodes, if necessary, we can assume that $v_k = k$ for $k = 1, \dots, p$. Now, the set of triangle inequalities (1), (2) is closed under switching, which implies that the set of SDCs is closed under switching as well. Accordingly, we can assume that our G2C inequality is simple.

To complete the proof, we show that the simple G2C inequalities (9) are implied by the triangle inequalities together with the simple disjunction

$$(x_{1p} = 0) \vee (x_{1p} = 1).$$

To this end, we consider two cases.

Case 1: $x_{1p} = 0$. In this case, we obtain the inequality (7) by summing together (a) the triangle inequalities

$$-x(i, i+1) - x(i+1, i+2) + x(i, i+2) \leq 0 \qquad (10)$$

for $i \in \{2, 4, \dots, p-1\} \cup \{p\}$; (b) the co-circuit inequality

$$\sum_{i=1}^{(p-5)/2} x(2i+1, 2i+3) + x(1,3) + x(p-2, p) - x(1, p) \leq (p-3)/2;$$

and (c) the inequality $2x_{1p} \leq 0$.

Case 2: $x_{1p} = 1$. In this case, we obtain the inequality (7) by summing together (a) the triangle inequalities (10) for $i = 1, 3, \dots, p-2$; (b) the triangle inequalities $x_{1p} + x_{12} + x_{2p} \leq 2$ and $x(1, p) + x(1, p-1) + x(p-1, p) \leq 2$; (c) the co-circuit inequality

$$\sum_{i=1}^{(p-3)/2} x(2i, 2i+2) + x(1,2) - x(1, p) - x(p-1, p) \leq (p-3)/2;$$

and (d) the inequality $-4x_{1p} \leq -4$. □

We remark that one of the inequalities mentioned in [1], which defines a facet of CUT_7, can be derived as a G2C inequality with $p = 7$. (That inequality is called a *parachute* inequality in [11].) In other words, the G2C inequalities with $p = 7$ are not completely new. On the other hand, the G2C inequalities with $p = 11, 15, \dots$ did not appear before in the literature.

To end this subsection, we use the collapsing operation to define some more families of inequalities. Let us say that an inequality is an *extended 2-circulant* (E2C) inequality if it is either a 2C inequality, or can be obtained from one via collapsing. We define *extended switched 2-circulant* (ES2C) and *extended generalised 2-circulant* (EG2C) inequalities analogously. One can check that EG2C inequalities can be written in the form (7), the only change being that the vertices $v_1, \dots, v_p$ are no longer required to be distinct.

$$\begin{tikzcd}
\text{2C} \arrow[r] \arrow[d] & \text{S2C} \arrow[r] \arrow[d] & \text{G2C} \arrow[d] \\
\text{E2C} \arrow[r] & \text{ES2C} \arrow[r] \arrow[d] & \text{EG2C} \arrow[d] \\
& \{0, \frac{1}{2}\} & \text{SDC}
\end{tikzcd}$$

Figure 4: Hierarchy of inequalities.

Figure 4 shows the resultant hierarchy of inequalities. An arrow from one class to another indicates that the former is a subset of the latter. (One can easily show that all inclusions are strict. We omit details for brevity.) This hierarchy will prove useful in the next section.

3.3. Facet proof

In this subsection, we will show that every G2C inequality defines a facet. We will need the following standard lemma.

Lemma 4. *[4, Lemma 2.5] Let $a^T x \leq b$ be a valid inequality for CUT_n, and let i, j be distinct nodes in K_n. If there is $W \subset V_n \setminus \{i, j\}$ such that the incidence vectors of $\delta(W)$, $\delta(W \cup \{i\})$, $\delta(W \cup \{j\})$ and $\delta(W \cup \{i, j\})$ satisfy $a^T x = b$, then $a_{ij} = 0$.*

Theorem 2. *Every G2C inequality defines a facet of CUT_n.*

Proof. If $p \equiv 1 \bmod 4$, then the G2C inequality is a switched 2C inequality, which is known to define a facet. So suppose that $p \equiv 3 \bmod 4$. By relabelling the nodes, if necessary, we can assume that $v_k = k$ for $k = 1, \dots, p$. Also, since the property of being facet-defining is preserved under switching, we can assume that $S^+ = \emptyset$. In other words, we can suppose that the G2C inequality is simple. Let us call edges of the form $\{i, i+1\}$ and $\{i, i+2\}$ 'outer' and 'inner', respectively.

We follow the proof strategy of [22, Theorem 4.1]. We say that a set $R \subset S$ is a 'root' if the incidence vector of the cut $\delta(R)$ satisfies the simple G2C inequality at equality. That is, R is a root if and only if the number of inner edges in $\delta(R)$ exceeds that of its outer edges by $(p-3)/2$. We also let $[m]$ stand for $\{1, 2, \dots, m\}$.

Assume an arbitrary but fixed inequality (9) and let F denote the face of CUT_n it defines, i.e., the convex hull of the incidence vectors of all it roots. As CUT_n is full-dimensional, F cannot coincide with CUT_n. Consider the following roots that show also the non-emptiness of F:

- $R_1 = \bigcup_{k\in[\frac{p-7}{4}]}\{4k+1, 4k+2\} \cup \{1, p-1, p\}$,
- $S_1 = \bigcup_{k\in[\frac{p-3}{4}]}\{4k-1, 4k\} \cup \{1, p\}$, and
- $T = \bigcup_{k\in[\frac{p-3}{4}]}\{4k, 4k+1\} \cup \{1\}$.

Simple counting shows that $\delta(R_1)$ contains $\frac{p-3}{2}$ outer and $p-3$ inner edges, while both $\delta(S_1)$ and $\delta(T)$ contain $\frac{p+1}{2}$ outer and $p-1$ inner edges. In fact, each of $\delta(S_1)$ and $\delta(T)$ contains all inner edges of the circulant except for $\{1, 3\}$ and $\{2, p\}$, respectively. Let us also observe that:

- R_1 contains $\{1, 2\}$ and $\{1, 3\}$ but not $\{1, p\}$ and $\{1, p-1\}$;
- S_1 contains $\{1, 2\}$ and $\{1, p-1\}$ but not $\{1, 3\}$ and $\{1, p\}$;
- and T contains all four edges incident to node 1.

Note also that switching node 1 to the opposite side of the cut in any of these three roots (i.e., deleting 1 from R_1 or S_1 or T) yields another root of (9) that we call the "{1}-switch" of the starting root. For example, deleting 1 from R_1 results in the following changes to the cut: (a) the outer edge $\{1, 2\}$ is removed, (b) the outer edge $\{1, p\}$ is added, (c) the inner edge $\{1, 3\}$ is removed, and (d) the inner edge $\{1, p-1\}$ is added. Thus, the total number of inner and outer edges in $\delta(R_1 \setminus \{1\})$ remains as in $\delta(R_1)$.

Let x^1 and $\hat{x}^1$ denote the incidence vectors of R_1 and its {1}-switch, y^1 and $\hat{y}^1$ the vectors of S_1 and its {1}-switch, and z^1 and $\hat{z}^1$ the vectors of T and its {1}-switch. For $i \in S$, one can define the roots R_i and S_i by 'shifting' by i positions the nodes in R_1 and T_1, thus obtaining also the corresponding $\{i\}$-switches and the vectors $x^i, \hat{x}^i, y^i$ and $\hat{y}^i$.

To show that F is a facet of CUT_n, we prove that any equality $a^T x = b$ satisfied by all roots in F has $a_e = \alpha$ for any inner e, $a_e = -\alpha$ for any outer e, and $a_e = 0$ for any other edge. That is, we show that any such inequality is a scalar multiple of (9).

As both x^i and $\hat{x}^i$ satisfy $a^T x = b$, we have $a^T x^i = a^T \hat{x}^i$. Thus,

$$a_{i(i+1)} + a_{i(i+2)} = a_{i(i-1)} + a_{i(i-2)}.$$

Similarly, we have $a^T y^i = a^T \hat{y}^i$, which yields

$$a_{i(i+1)} + a_{i(i-2)} = a_{i(i-1)} + a_{i(i+2)}.$$

Adding the two shows that $a_{i(i+1)} = a_{i(i-1)}$ and, therefore, $a_{i(i+2)} = a_{i(i-2)}$. This implies (by the rotational symmetry of the circulant) that $a_e = \alpha$ for any inner e and $a_e = \beta$ for any outer e. Moreover, $a^T z^1 = a^T \hat{z}^1$, which yields

$$a_{12} + a_{13} + a_{1p} + a_{1(p-1)} = 0.$$

Combined with the above, this shows that $\beta = -\alpha$.

It remains to show that $a_e = 0$ for any other edge $e = \{i, j\}$, using Lemma 4. If neither of i and j belongs to S, any subset of S that is a root can play the role of W in Lemma 4. If only i belongs to S, one can set W to $R_i \setminus \{i\}$. Finally, if both i and j belong to S, we can assume without loss of generality that $i = 1$. This in turn implies that $j \in \{4, 5, \ldots, p-3, p-2\}$, given that $\{i, j\}$ is not an edge of the circulant. If j is congruent to 0 or 3 mod 4, then j is not in R_1, and therefore we can set W to $R_1 \setminus \{1\}$; otherwise, j is not in S_1, and we can set W to $S_1 \setminus \{1\}$. □

4. The New Separation Algorithm

When considering separation, it helps to refer once again to Figure 4. Recall that the algorithms in [19] and [20] separate over the $\{0, \frac{1}{2}\}$-cuts and the SDCs, respectively. One can also check that the separation algorithm in [15] separates over the ES2C inequalities.

We now present an $O(n^5)$ separation algorithm for the EG2C inequalities. The algorithm is based on the following definition and lemma.

Definition 2. *Given any ordered triple (i, j, k) of nodes in V_n, define*

$$\begin{aligned}
\Delta^{++}(i, j, k) &= 3 - x_{ij} - x_{jk} - 2x_{ik} \\
\Delta^{+-}(i, j, k) &= -x_{ij} + x_{jk} + 2x_{ik} \\
\Delta^{-+}(i, j, k) &= -x_{ij} + x_{jk} + 2x_{ik} \\
\Delta^{--}(i, j, k) &= 1 + x_{ij} - x_{jk} - 2x_{ik}.
\end{aligned}$$

Note that these quantities are the slacks of the inequalities (3)-(6).

Lemma 5. *The G2C inequalities (7) can be written as*

$$\sum_{i\in S^{++}} \Delta^{++}(v_i, v_{i+1}, v_{i+2}) + \sum_{i\in S^{+-}} \Delta^{+-}(v_i, v_{i+1}, v_{i+2}) + \sum_{i\in S^{-+}} \Delta^{-+}(v_i, v_{i+1}, v_{i+2}) + \sum_{i\in S^{--}} \Delta^{--}(v_i, v_{i+1}, v_{i+2}) \geq 3,$$

where indices are again taken mod p.

Proof. Multiply the inequality (7) by minus two, add $3|S^{++}| + |S^{--}|$ to both sides, and re-arrange the left-hand side. □

We remark that the EG2C inequalities can also be written as in the above lemma, the only change being that the vertices $v_1, \ldots, v_p$ are no longer required to be distinct.

Now, assume that we have been given a point $x^* \in [0,1]^{E_n}$ that we wish to separate. We construct an auxiliary graph, called $\tilde{G}$, as follows. For each ordered pair (i, j) of nodes in V, we insert two nodes into $\tilde{G}$, labelled "$(i, j)^+$" and "$(i, j)^-$". For each ordered triple (i, j, k) of nodes in V, we include edges in $\tilde{G}$ as follows:

- an edge between node $(i, j)^+$ and node $(j, k)^+$, with "weight" of $\Delta^{++}(i, j, k)$ (evaluated at x^*) and a "charge" of 3;
- an edge between node $(i, j)^+$ and node $(j, k)^-$, with weight $\Delta^{+-}(i, j, k)$ and charge 0;
- an edge between node $(i, j)^-$ and node $(j, k)^+$, with weight $\Delta^{-+}(i, j, k)$ and charge 0;
- an edge between node $(i, j)^-$ and node $(j, k)^-$, with weight $\Delta^{--}(i, j, k)$ and charge 1.

It then follows from Lemma 5 that, if x^* violates a G2C inequality, there is a circuit in $\tilde{G}$ that (a) has total weight less than 3, and (b) has a total "charge" congruent to 3 mod 4. It can also be shown, using the remark following Lemma 5, that x^* violates an EG2C inequality if and only if there is such a circuit in $\tilde{G}$.

To detect whether such a circuit exists, we use an idea from [4, 15]. We construct another graph, called G^+, which is four times larger than $\tilde{G}$. For each pair (i, j), and for $c = 0, \ldots, 3$, we have a node labelled "$(i, j)^+_c$" and a node labelled "$(i, j)^-_c$". The index c represents the cumulative "charge", and is taken modulo 4. For each triple (i, j, k), and for $c = 0, \ldots, 3$, we have an edge between $(i, j)^+_c$ and $(j, k)^+_{c+3}$ with weight $\Delta^{++}(i, j, k)$, an edge between $(i, j)^+_c$ to $(j, k)^-_c$ with weight $\Delta^{++}(i, j, k)$, and so on.

We can now solve the separation problem by solving $O(n^2)$ shortest-path problems in G^+. Specifically, for each pair (i, j), we find the minimum-weight path from $(i, j)^+_0$ to $(i, j)^+_3$, and the minimum-weight path from $(i, j)^-_0$ to $(i, j)^-_3$. If the path has a total weight less than 3, we have found a violated inequality.

Note that D^+ has $O(n^2)$ nodes and $O(n^3)$ arcs, and it can be constructed in $O(n^3)$ time. Moreover, under the assumption that our fractional point x^* satisfies all of the weakened triangle inequalities, each arc in A^+ will have non-negative weight. Thus, we can use Dijkstra's algorithm to solve the shortest-path problems. Using the Fibonacci heap variant of Dijkstra's algorithm [12], one can solve each shortest-path problem in $O(n^3)$ time. This leads to a total running time of $O(n^5)$.

A running time of $O(n^5)$ is still rather high. The algorithm can be made faster by exploiting the sparsity of x^*. Define the edge set $F = \{e \in E : 0 < x^*_e < 1\}$. (The "F" stands for "fractional".) Proposition 3 implies that, if x^* satisfies all triangle inequalities, then an EG2C inequality cannot be violated unless $(v_i, v_{i+1}) \in F$ for $i = 1, \ldots, p$.

To exploit this fact, we assume that x^* satisfies all triangle inequalities. We then only include the nodes $(i, j)^+_c$ and $(i, j)^-_c$ in G^+ if x^*_{ij} is fractional. This reduces the number of nodes and edges in G^+ to $O(|F|)$ and $O(n|F|)$, respectively. Each shortest-path computation then takes only $O(n|F|)$ time, and the number of shortest-path calls reduces to $O(|F|)$. The total time reduces to $O(n|F|^2)$.

Although the running time of $O(n|F|^2)$ is still rather high, note that the algorithm can generate several violated inequalities in a single call.

We remark that our auxiliary graph G^+ has half as many nodes and edges as the one in [15], despite the fact that it separates over a larger family of inequalities.

5. Computational Results

To explore the potential of EG2C inequalities, we modified the cutting-plane algorithm described in [15], and ran it on the same instances. The test set is composed of two sets of fully dense instances, called "MC_A" and "MC_B". For the instances in MC_A, each edge weight is a random integer uniformly selected from $\{1, \ldots, 10\}$. For the instances in MC_B, each edge weight is set to either $+1$ or -1, with equal probability. Each set contains ten instances for each value of n in $\{35, 45, 55\}$.

The code was written in C and calls on the callable library of `CPLEX (v.12.8)`. We use primal simplex to solve the initial LP and dual simplex to re-optimise after adding cutting planes. The experiments were run on an Intel i5 processor at 3.40GHz × 4, under Ubuntu 18.04, with 8GB of RAM.

We considered three versions of the cutting-plane algorithm. In version (a), we separate triangle inequalities alone. In version (b), we separate over the ES2C inequalities, using the algorithm in [15], when triangle separation fails. In version (c), we separate over the (more general) EG2C inequalities, using the algorithm in this paper, when triangle separation fails.

For each instance and each version of the cutting-plane algorithm, we stored the total computing time and the upper bound obtained. We also calculated the gap

	$\lvert V\rvert$	(a)	(b)	(c)
	35	0.72	1120	119
MC_A	45	0.97	3760	449
	55	2.34	10383	1320
	35	0.34	425	94
MC_B	45	1.34	4965	1501
	55	5.05	24076	6161

Table 1: Average running times (in seconds) for three versions of the cutting-plane algorithm.

	$\lvert V\rvert$	(a)	(b)	(c)
	35	15.37	3.83	3.83
MC_A	45	17.83	5.63	5.63
	55	21.33	7.35	7.35
	35	33.90	0.00	0.00
MC_B	45	55.74	0.37	0.37
	55	78.89	3.89	3.89

Table 2: Average percentage integrality gaps for three versions of the cutting-plane algorithm.

between each upper bound and the optimum, expressed as a percentage of the optimum.

Table 1 shows the average computing times, in seconds. Each row corresponds to a batch of ten instances. It is apparent that the cutting-plane algorithm converges much more quickly when our separation algorithm is used rather than the one in [15], despite the fact that our separation algorithm separates over a more general family of inequalities than the one in [15]. Table 2 shows the average percentage gaps for the same instances. We observe that the ES2C and EG2C inequalities close an impressive proportion of the integrality gap, especially for the MC_B instances. Remarkably though, the numbers in the last two columns are identical. In fact, versions (b) and (c) of the cutting-plane algorithm produced identical upper bounds for every one of our 60 instances. In an attempt to understand this phenomenon better, we inspected the primal and dual LP solutions obtained when the cutting-plane algorithm terminates. Recall that an inequality with zero slack is called *binding*. It turned out that, regardless of whether version (b) or (c) was used, the majority of the binding inequalities were ES2C inequalities with $p = 5$ (which is the smallest value that p can take). There were some other binding ES2C and EG2C inequalities, but they all had zero dual price.

Another strange phenomenon is that, for almost all instances, the number of binding EG2Cs when version (c) was used was significantly smaller than the number of binding ES2Cs when version (b) was used. We suspect that this odd behaviour is due to the fact that all of our instances were fully dense.

An interesting topic for future work is the integration of EG2C inequalities (perhaps together with other known inequalities) in a branch-and-cut framework. Another interesting topic is to determine whether any other arrows should be added to Figure 4. In particular, can EG2C inequalities be derived as $\{0, \frac{1}{2}\}$-cuts, and can $\{0, \frac{1}{2}\}$-cuts be derived as SDCs?

References

[1] P. Assouad (1984) Sur les inégalités valides dans L^1. *Eur. J. Combinatorics*, 5, 99–112.

[2] E. Balas, S. Ceria & G. Cornuéjols (1993) A lift-and-project cutting plane algorithm for mixed 0-1 programs. *Math. Program.*, 58, 295–324.

[3] F. Barahona, M. Jünger & G. Reinelt (1989) Experiments in quadratic 0-1 programming. *Math. Program.*, 44, 127–137.

[4] F. Barahona & A.R. Mahjoub (1986) On the cut polytope. *Math. Program.*, 36, 157–173.

[5] T. Bonato, M. Jünger, G. Reinelt & G. Rinaldi (2014) Lifting and separation procedures for the cut polytope. *Math. Program.*, 146, 351–378.

[6] E. Boros & P.L. Hammer (1993) Cut-polytopes, Boolean quadric polytopes and nonnegative quadratic pseudo-Boolean functions. *Math. Oper. Res.*, 18, 245–253.

[7] A. Caprara & M. Fischetti (1996) $\{0, 1/2\}$-Chvátal-Gomory cuts. *Math. Program.*, 74, 221–235.

[8] M. Conforti, G. Cornuéjols & G. Zambelli (2014) *Integer Programming*. Cham, Switzerland: Springer.

[9] C. De Simone, M.M. Deza & M. Laurent (1994) Collapsing and lifting for the cut cone. *Discr. Math.*, 127, 105–130.

[10] M.M. Deza & M. Laurent (1992) Facets for the cut cone I. *Math. Program.*, 56, 121–160.

[11] M.M. Deza & M. Laurent (1997) *Geometry of Cuts and Metrics*. Berlin: Springer.

[12] M.L. Fredman & R.E. Tarjan (1987) Fibonacci heaps and their uses in improved network optimization algorithms. *J. ACM*, 34, 596–615.

[13] M.R. Garey, D.S. Johnson & L.J. Stockmeyer (1976) Some simplified $\mathcal{NP}$-complete graph problems. *Theoret. Comput. Sci.*, 1, 237–267.

[14] A.M.H. Gerards (1985) Testing the odd bicycle wheel inequalities for the bipartite subgraph polytope. *Math. Oper. Res.*, 10, 359–360.

[15] K. Kaparis & A.N. Letchford (2018) On the 2-circulant inequalities for the max-cut problem. *Oper. Res. Lett.*, 46, 443–447.

[16] M. Laurent (1997) Max-cut problem. In M. Dell'Amico, F. Maffoli & S. Martello (eds.) *Annotated Bibliographies in Combinatorial Optimization*, pp. 241–259. Chichester: Wiley.

[17] M. Laurent & S. Poljak (1995) On a positive semidefinite relaxation of the cut polytope. *Lin. Alg. Appl.*, 223/224, 439–461.

[18] M. Laurent & S. Poljak (1996) Gap inequalities for the cut polytope. *Eur. J. Combinatorics*, 17, 233–254.

[19] A.N. Letchford (2001) On disjunctive cuts for combinatorial optimization. *J. Combin. Optim.*, 5, 299–315.

[20] A.N. Letchford & M.M. Sørensen (2014) A new separation algorithm for the Boolean quadric and cut polytopes. *Discr. Optim.*, 14, 61–71.

[21] L. Palagi, V. Piccialli, F. Rendl, G. Rinaldi & A. Wiegele (2012) Computational approaches to max-cut. In M.F. Anjos & J.B. Lasserre (eds.) *Handbook on Semidefinite, Conic and Polynomial Optimization*, pp. 821–847. Boston, MA: Springer US.

[22] S. Poljak & D. Turzik (1992) Max-cut in circulant graphs. *Discr. Math.*, 108, 379–392.

[23] F. Rendl, G. Rinaldi & A. Wiegele (2010) Solving max-cut to optimality by intersecting semidefinite and polyhedral relaxations. *Math. Program.*, 121, 307–335.