The Boolean Quadric Polytope

Adam N. Letchford

Abstract When developing an exact algorithm for a combinatorial optimisation problem, it often helps to have a good understanding of certain *polyhedra* associated with that problem. In the case of quadratic unconstrained Boolean optimisation, the polyhedron in question is called the *Boolean quadric polytope*. This chapter gives a brief introduction to polyhedral theory, reviews the literature on the Boolean quadric polytope and related polyhedra, and explains the algorithmic implications.

1 Introduction

It has been known for some time that *Quadratic unconstrained Boolean optimisation* (QUBO) is equivalent to another well-known combinatorial optimisation problem, known as the *max-cut* problem [6, 14, 53]. The max-cut problem has been proven to be "strongly $\mathcal{NP}$-hard" [26], and therefore the same holds for QUBO. Rather than explaining strong $\mathcal{NP}$-hardness in detail, let us just say that it makes it unlikely that an algorithm can be developed which solves all QUBO instances quickly.

The situation however is far from hopeless. Indeed, for many specific $\mathcal{NP}$-hard problems, algorithms have been developed that can solve many instances of interest to proven optimality (or near-optimality) in reasonable computing times. Many of these algorithms use a method known as *branch-and-cut* (see, e.g., [10, 51, 54]). Branch-and-cut is an enumerative scheme, in which a "tree" of subproblems is explored, and each subproblem is a *linear program* (LP).

One of the keys to designing a successful branch-and-cut algorithm for a given problem is to gain an understanding of certain *polyhedra* associated with that problem (e.g., [1–3, 12]). In the case of QUBO, the polyhedron in question is called the *Boolean quadric polytope* (e.g., [9, 14, 53]).

Adam N. Letchford
Department of Management Science, Lancaster University, Lancaster LA1 4YX, United Kingdom.
e-mail: a.n.letchford@lancaster.ac.uk

This chapter gives a brief introduction to polyhedral theory, a detailed survey of known results on the Boolean quadric polytope, and a brief discussion of algorithmic implications. The structure of the chapter is as follows. The basics of polyhedral theory are recalled in Section 2. In Section 3, we define the Boolean quadric polytope and mention some of its fundamental properties. In Section 4, we survey some of the known valid inequalities for the Boolean quadric polytope. In Section 5, we review some connections between the Boolean quadric polytope and some other important polytopes. In Section 6, we mention some other related convex sets. In Section 7, we look at the algorithmic implications. Finally, concluding remarks are made in Section 8.

We use the following conventions and notation throughout the chapter. Given a positive integer n, we sometimes write V_n for $\{1, \ldots, n\}$. We K_n denote the complete graph on the vertex set V_n, and let E_n denote its edge set. Given a vector $\mathbf{v} \in \mathbb{R}^n$, we let $\sigma(\mathbf{v})$ denote $\sum_{i=1}^n v_i$. All matrices are real. Given two matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m \times n}$, we write $\mathbf{A} \bullet \mathbf{B}$ for the (Frobenius) inner product

$$\sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} b_{ij} = \mathrm{Tr}(\mathbf{A}^T \mathbf{B}).$$

Given a positive integer k, we let $\mathcal{S}_+^k$ denote the set of positive semidefinite (psd) matrices of order k. We recall that a symmetric matrix $\mathbf{M}$ of order k is psd if and only if all of its eigenvalues are non-negative, or, equivalently, $\mathbf{v}^T \mathbf{M} \mathbf{v} \geq 0$ for all vectors $\mathbf{v} \in \mathbb{R}^k$.

2 Elementary Polyhedral Theory

This section draws on material from [33, 52].

Suppose that $\mathbf{x}^1, \ldots, \mathbf{x}^k \in \mathbb{R}^n$ are (column) vectors and $\lambda_1, \ldots, \lambda_k$ are scalars. A vector of the form $\lambda_1 \mathbf{x}^1 + \cdots + \lambda_k \mathbf{x}^k$ is called a *linear combination* of $\mathbf{x}^1, \ldots, \mathbf{x}^k$. It is called a *conical* combination if $\lambda_1, \ldots, \lambda_k$ are non-negative, an *affine* combination if $\sum_{i=1}^k \lambda_i = 1$, and a *convex* combination if it is both conical and affine. Given some non-empty set $S \subset \mathbb{R}^n$, the *convex hull* of S is the set of all convex combinations of the vectors in S. The linear, affine and conical hulls are defined analogously. We will let $\mathrm{conv}(S)$ denote the convex hull of S.

A set $S \subseteq \mathbb{R}^n$ is called *convex* if $\lambda \mathbf{x}^1 + (1 - \lambda)\mathbf{x}^2 \in S$ holds for all $\mathbf{x}^1, \mathbf{x}^2 \in S$ and all $\lambda \in (0, 1)$. A convex set P is called a *polyhedron* if there exists a non-negative integer m, a matrix $\mathbf{A} \in \mathbb{Z}^{mn}$ and a vector $\mathbf{b} \in \mathbb{Z}^m$ such that

$$P = \left\{ \mathbf{x} \in \mathbb{R}^n : \mathbf{A}\mathbf{x} \leq \mathbf{b} \right\}.$$

A polyhedron which is bounded (i.e., not of infinite volume) is called a *polytope*. A famous theorem of Weyl [68] states that a set $P \subset \mathbb{R}^n$ is a polytope if and only if it is the convex hull of a finite number of points.

A point $\mathbf{x} \in P$ is called an *extreme point* of P if it is not a convex combination of other points in P. Every polytope is the convex hull of its extreme points.

A set of vectors is called *affinely independent* if no member of the set is an affine combination of the others. The *dimension* of a polyhedron P, denoted by $dim(P)$, is the maximum number of affinely independent vectors in P, minus one. Note that $dim(P) \leq n$. If equality holds, P is said to be *full-dimensional*.

A linear inequality $\mathbf{a}^T\mathbf{x} \leq a_0$ is *valid* for a polyhedron P if it is satisfied by every point in P. The set

$$F = P \cap \left\{\mathbf{x} \in \mathbb{R}^n : \mathbf{a}^T\mathbf{x} \leq a_0\right\}$$

is called the *face* of P *induced* by the given inequality. Note that F is itself a polyhedron. The face F is called a *facet* of P if $dim(F) = dim(P) - 1$.

Example: Suppose that S contains the following four points in $\mathbb{R}^3$:

$$\mathbf{x}^1 = \begin{pmatrix}0\\0\\1\end{pmatrix}, \quad \mathbf{x}^2 = \begin{pmatrix}0\\2\\1\end{pmatrix}, \quad \mathbf{x}^3 = \begin{pmatrix}2\\0\\1\end{pmatrix} \quad \mathbf{x}^4 = \begin{pmatrix}1\\1\\1\end{pmatrix}.$$

One can check that:

- the linear hull of S is $\mathbb{R}^3$ itself;
- the conical hull is $\left\{\mathbf{x} \in \mathbb{R}^3 : x_1 \geq 0, x_2 \geq 0, x_1 + x_2 \leq 2x_3\right\}$;
- the affine hull is $\left\{\mathbf{x} \in \mathbb{R}^3 : x_3 = 1\right\}$;
- $\text{conv}(S) = \left\{\mathbf{x} \in \mathbb{R}^3 : x_1 \geq 0, x_2 \geq 0, x_1 + x_2 \leq 2, x_3 = 1\right\}$.

Now let $P = \text{conv}(S)$. One can check that (a) P is a polytope, (b) $\dim(P) = 2$, (c) the extreme points of P are $\mathbf{x}^1, \ldots, \mathbf{x}^3$ and (d) P has three facets, induced by the inequalities $x_1 \geq 0, x_2 \geq 0$ and $x_1 + x_2 \leq 2$. □

We now explain the connection between polyhedra and combinatorial optimisation. Suppose we can formulate our optimisation problem as an *integer linear program* (ILP) of the form

$$\max\left\{\mathbf{cx} : \mathbf{Ax} \leq \mathbf{b}, \mathbf{x} \in \mathbb{Z}^n_+\right\}. \tag{1}$$

Replacing the condition $\mathbf{x} \in \mathbb{Z}^n_+$ with the weaker condition $\mathbf{x} \in \mathbb{R}^n_+$, we obtain the so-called *continuous relaxation* of the ILP. The continuous relaxation is an LP, which is likely to be easy to solve. Let $\mathbf{x}^*$ be a (basic) optimal solution to the continuous relaxation. If $\mathbf{x}^*$ is integral, we have solved the ILP. Otherwise we have to do more work, and this is where polyhedra come into play.

The feasible region of the continuous relaxation is the polyhedron

$$P = \left\{\mathbf{x} \in \mathbb{R}^n_+ : \mathbf{Ax} \leq \mathbf{b}\right\},$$

and the set of feasible solutions to the ILP is $S = P \cap \mathbb{Z}^n_+$. The convex hull of S is also a polyhedron, called the *integral hull* of P. We will denote it by P_I. By definition, we have $P_I \subseteq P$. Also, if $\mathbf{x}^*$ is not integral, then P_I is strictly contained in P, and there must exist a linear inequality that is valid for P_I but violated by $\mathbf{x}^*$. Such an inequality is called a *cutting plane*.

Example: Consider the ILP

$$\begin{array}{rr} \max & x_1 + x_2 \\ \text{s.t.} & 4x_1 + 2x_2 \le 15 \\ & 2x_1 - 2x_2 \le 5 \\ & -2x_1 + 2x_2 \le 3 \\ & -6x_1 - 10x_2 \le -15 \\ & 2x_2 \le 5 \\ & \mathbf{x} \in \mathbb{R}^2_+. \end{array}$$

On the left of Figure 2, we show the polyhedron P. Points with integer coordinates are represented by small circles. On the right of Figure 2, the points in S are represented as larger circles. One can check that there are two optimal solutions to the ILP, $\binom{2}{2}$ and $\binom{3}{1}$, each with profit 4. The solution to the continuous relaxation, on the other hand, is $\mathbf{x}^* = \binom{2.5}{2.5}$, giving an upper bound of 5. On the left of Figure 2, we show the integral hull P_I. Finally, on the right of Figure 2, we show both P and P_I, together with a possible cutting plane, represented by a dashed line. □

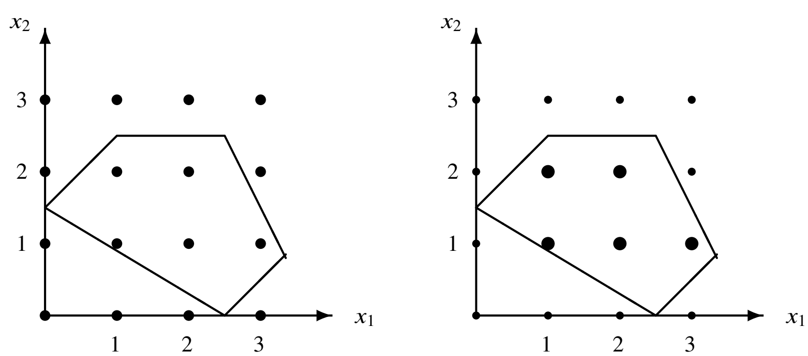

Fig. 1 Polyhedron P (left) and set S of integer solutions (right).

For simplicity and brevity, we assume from now on that P_I (and therefore also P) is a full-dimensional polytope. Under this assumption, the strongest possible cutting planes for a given ILP are those that induce facets of P_I.

At this point we should mention some negative results from Karp & Papadimitriou [39]. They showed that, if a combinatorial optimisation problem is $\mathcal{NP}$-hard, then, regardless of how it is formulated as an ILP, it is $\mathcal{NP}$-hard to check if a given linear

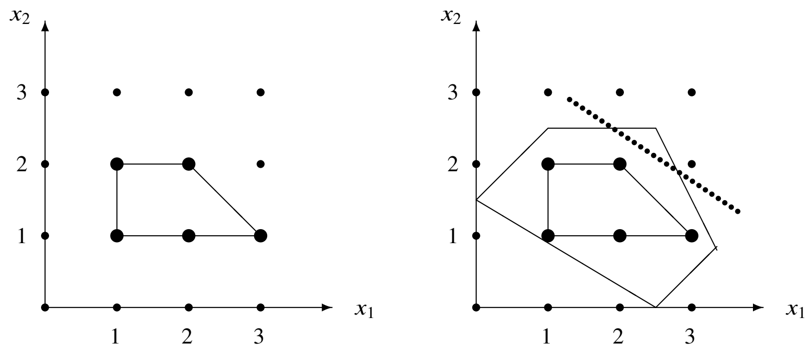

Fig. 2 Polyhedron P_I (left) and a possible cutting plane (right).

inequality is valid for the associated polytope P_I. They also show that it is $\mathcal{NP}$-hard to check if a given inequality induces a facet of P_I.

Although the above-mentioned results may appear discouraging, there is also good news: for many important combinatorial optimisation problems (such as the knapsack problem, the travelling salesman problem, the stable set problem, and QUBO itself), researchers have discovered several large families of facet-inducing inequalities (see, e.g., [1–3, 12, 33, 52]). These inequalities can be used as cutting planes in branch-and-cut algorithms.

3 The Boolean Quadric Polytope

Now consider a QUBO instance of the form:

$$\begin{aligned} \max \quad & \mathbf{x}^T \mathbf{Q} \mathbf{x} \\ \text{s.t. } & \mathbf{x} \in \{0,1\}^n, \end{aligned}$$

where, without loss of generality, we assume that $\mathbf{Q}$ is symmetric. Glover & Woolsey [28] proposed to replace each quadratic term $x_i x_j$ with a new binary variable y_{ij}. This allows one to formulate QUBO as the following 0-1 LP:

$$\max \sum_{i \in V_n} q_{ii} x_i + 2 \sum_{\{i,j\} \in E_n} q_{ij} y_{ij} \tag{2}$$

$$\text{s.t.} \quad y_{ij} \le x_i \qquad (\{i,j\} \in E_n) \tag{3}$$

$$y_{ij} \le x_j \qquad (\{i,j\} \in E_n) \tag{4}$$

$$x_i + x_j \le y_{ij} + 1 \qquad (\{i,j\} \in E_n) \tag{5}$$

$$\mathbf{x} \in \{0,1\}^n \tag{6}$$

$$\mathbf{y} \in \{0,1\}^{\binom{n}{2}}. \tag{7}$$

For a given $n \geq 2$, the convex hull of pairs $(\mathbf{x}, \mathbf{y})$ satisfying (3)–(7) is called the *Boolean quadric polytope of order n* and denoted by BQP_n [14, 53]. (Some authors call it the *correlation polytope* instead; see, e.g., [22, 55].)

To make this clear, consider the case $n = 2$. To obtain a feasible solution to the 0-1 LP, we require:

$$\begin{pmatrix} x_1 \\ x_2 \\ y_{12} \end{pmatrix} \in \left\{ \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \right\}.$$

One can check that the four points in question are affinely independent. Thus, BQP_2 is a tetrahedron, as shown in Figure 3. Its facets are induced by the inequalities $y_{12} \leq x_1$, $y_{12} \leq x_2$, $y_{12} \geq x_1 + x_2 - 1$ and $y_{12} \geq 0$.

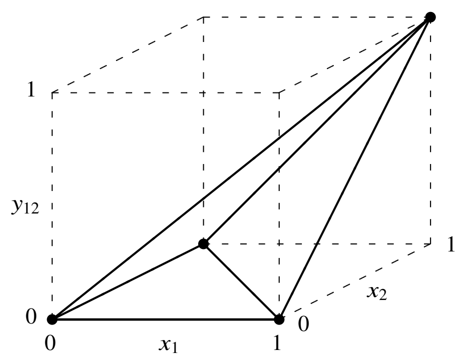

Fig. 3 The Boolean quadric polytope of order 2.

Padberg [53] proved that BQP_n is full-dimensional and that the inequalities (3)–(5), along with the non-negativity inequalities $y_{ij} \geq 0$, always induce facets. He also derived some additional inequalities, which we review in Section 4.

The Boolean quadric polytope has some remarkable properties. For one thing, every extreme point of BQP_n is adjacent to every other one [60]. Moreover, BQP_n has a high degree of *symmetry*. In particular, BQP_n is invariant under two transformations, called *permutation* and *switching* [22, 53, 55]. These are defined as follows.

Definition 1 (Permutation) Let $\pi : V_n \mapsto V_n$ be an arbitrary permutation. Consider the linear transformation $\phi^\pi : \mathbb{R}^{n+\binom{n}{2}} \mapsto \mathbb{R}^{n+\binom{n}{2}}$ that:

- replaces x_i with $x_{\pi(i)}$ for all $i \in V_n$,
- replaces y_{ij} with $y_{\pi(i),\pi(j)}$ for all $\{i, j\} \in E_n$.

By abuse of terminology, we call this transformation itself a "permutation".

Definition 2 (Switching) For an arbitrary set $S \subset V_n$, let $\psi^S : \mathbb{R}^{n+\binom{n}{2}} \mapsto \mathbb{R}^{n+\binom{n}{2}}$ be the affine transformation that:

- replaces x_i with $1 - x_i$ for all $i \in S$,
- replaces y_{ij} with $x_i - y_{ij}$ for all $i \in \{1, \ldots, n\} \setminus S$ and all $j \in S$,

- replaces y_{ij} with $1 - x_i - x_j + y_{ij}$ for all $\{i, j\} \subset S$,
- leaves all other x_i and y_{ij} variables unchanged.

Applying the transformation ψ^S is called "switching" (on S).

It is fairly easy to show that BQP_n is invariant under permutation. (That is, for any n and any permutation π of $\{1, \ldots, n\}$, we have $\phi^\pi(\text{BQP}_n) = \text{BQP}_n$.) To make this chapter self-contained, we now show that the same holds for switching:

Proposition 1 BQP_n *is invariant under switching. That is, for any n and any $S \subset V_n$,* $\psi^S\big(\text{BQP}_n\big) = \text{BQP}_n$.

Proof Let $(\bar{\mathbf{x}}, \bar{\mathbf{y}})$ be an extreme point of BQP_n. By definition, we have $\bar{x}_i \in \{0, 1\}$ for $i \in V_n$ and $\bar{y}_{ij} = \bar{x}_i \bar{x}_j$ for $\{i, j\} \in E_n$. Now let $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) = \psi^S(\bar{\mathbf{x}}, \bar{\mathbf{y}})$. From the definition of switching, we have $\tilde{x}_i \in \{0, 1\}$ for $i \in V_n$ and $\tilde{y}_{ij} = \tilde{x}_i \tilde{x}_j$ for $\{i, j\} \in E_n$. Thus, $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$ is also an extreme point of BQP_n. This shows that every extreme point of $\psi^S\big(\text{BQP}_n\big)$ is an extreme point of BQP_n. A similar argument shows that every extreme point of BQP_n is an extreme point of $\psi^S\big(\text{BQP}_n\big)$. Now, recall that BQP_n is a polytope. Given that switching is an affine transformation, $\psi^S\big(\text{BQP}_n\big)$ must be a polytope as well. Thus, BQP_n and $\psi^S\big(\text{BQP}_n\big)$ are polytopes with the same extreme points, and are therefore equal. □

The permutation and switching transformations are very useful, because they enable one to convert valid linear inequalities for BQP_n into other valid linear inequalities that induce faces of the same dimension. For example, if we take the inequality $y_{ij} \geq 0$ and switch on $\{i\}$ or $\{j\}$, we obtain the inequalities $y_{ij} \leq x_j$ and $y_{ij} \leq x_i$, respectively. If we switch on $\{i, j\}$ instead, we obtain the inequality $y_{ij} \geq x_i + x_j - 1$.

We remark that switching on S and then switching on T is equivalent to switching on the set $(S \cup T) \setminus (S \cap T)$. Thus, given any valid (or facet-inducing) inequality for BQP_n, we can obtain up to $2^n - 1$ other valid (or facet-inducing) inequalities by switching.

4 Some More Valid Inequalities

In this section, we review some additional valid inequalities for BQP_n.

Padberg [53] derived three additional families of inequalities. The first are the following *triangle* inequalities:

$$x_i + x_j + x_k \leq y_{ij} + y_{ik} + y_{jk} + 1 \quad \big(\{i, j, k\} \subseteq V_n\big) \tag{8}$$

$$y_{ij} + y_{ik} \leq x_i + y_{jk} \quad \big(i \in V_n,\ \{j, k\} \subseteq V_n \setminus \{i\}\big). \tag{9}$$

To see how these might be useful as cutting planes, observe that fractional points with $x_i = x_j = x_k = 1/2$ and $y_{ij} = y_{ik} = y_{jk} = 0$ satisfy (3)-(5), but violate (8). Similarly, fractional points with $x_i = x_j = x_k = y_{ij} = y_{ik} = 1/2$ and $y_{jk} = 0$ satisfy (3)-(5), but violate (9). Note that (9) can be obtained from (8) by switching on $\{i\}$.

Padberg's second family are called *clique* inequalities. The easiest way to derive them is to note that, given any integer s, we have $s(s+1) \geq 0$. Thus, for any $S \subseteq V_n$ and any integer s, we have

$$\left(\sum_{i \in S} x_i - s\right)\left(\sum_{i \in S} x_i - s - 1\right) \geq 0.$$

Expanding this and re-arranging yields

$$(2s+1)\sum_{i \in S} x_i \; - \; \sum_{i \in S} x_i^2 \; \leq \; 2 \sum_{\{i,j\} \subseteq S} x_i x_j + s(s+1).$$

Linearising and dividing by two yields the clique inequalities:

$$s \sum_{i \in S} x_i \leq \sum_{\{i,j\} \subseteq S} y_{ij} + \binom{s+1}{2} \qquad \big(S \subseteq V_n,\ s = 0, \ldots, |S|-1\big). \tag{10}$$

Padberg showed that these induce facets when $|S| \geq 3$ and $1 \leq s \leq |S| - 2$.

Note that the clique inequalities (10) reduce to the triangle inequalities (8) when $|S| = 3$ and $s = 1$. Moreover, the inequalities (5) can be regarded as "degenerate" clique inequalities with $|S| = 2$ and $s = 1$. In a similar way, the non-negativity inequalities $y_{ij} \geq 0$ can be regarded as "degenerate" clique inequalities with $|S| = 2$ and $s = 0$.

Padberg's last family are called *cut* inequalities. They can be derived from the fact that, for any disjoint sets $S, T \subset V_n$, we have

$$\left(\sum_{i \in S} x_i - \sum_{i \in T} x_i\right)\left(\sum_{i \in S} x_i - \sum_{i \in T} x_i - 1\right) \geq 0.$$

They take the form:

$$\sum_{i \in S, j \in T} y_{ij} \leq \sum_{i \in T} x_i + \sum_{\{i,j\} \subseteq S} y_{ij} + \sum_{\{i,j\} \subseteq T} y_{ij} \qquad \big(S, T \subseteq V_n, S \cap T = \emptyset\big). \tag{11}$$

They induce facets when $|S| \geq 1$ and $|T| \geq 2$.

Note that the cut inequalities (11) reduce to the triangle inequalities (9) when $|S| = 2$ and $|T| = 1$. Moreover, the inequalities (3) and (4) can be regarded as "degenerate" cut inequalities with $|S| = |T| = 1$.

Next, we observe that the arguments for proving the validity of the clique and cut inequalities can be easily generalised. Indeed, for any disjoint sets $S, T \subset V_n$ and any $s \in \mathbb{Z}$, we have

$$\left(\sum_{i \in S} x_i - \sum_{i \in T} x_i - s\right)\left(\sum_{i \in S} x_i - \sum_{i \in T} x_i - s - 1\right) \geq 0.$$

Expanding and linearising yields

$$s \sum_{i \in S} x_i + \sum_{i \in S, j \in T} y_{ij} \leq (s+1) \sum_{i \in T} x_i + \sum_{\{i,j\} \subseteq S} y_{ij} + \sum_{\{i,j\} \subseteq T} y_{ij} + \binom{s+1}{2}. \qquad (12)$$

These inequalities, which include all those mentioned so far, have been rediscovered many times (e.g., [9, 15, 22, 69]). They define facets when $|S| + |T| \geq 3$ and $1 - |T| \leq s \leq |S| - 2$. We remark that they can also be derived by taking the clique inequality (10), and switching on T.

An even larger family of valid inequalities was found by Boros & Hammer [9]. Take an arbitrary vector $\mathbf{v} \in \mathbb{Z}^n$ and integer s, and consider the quadratic inequality $(\mathbf{v}^T\mathbf{x} - s)(\mathbf{v}^T\mathbf{x} - s - 1) \geq 0$. Expanding and linearising yields:

$$\sum_{i \in V_n} v_i (2s + 1 - v_i) x_i \leq 2 \sum_{1 \leq i < j \leq n} v_i v_j y_{ij} + s(s+1). \qquad (13)$$

Although the Boros-Hammer inequalities are infinite in number, it is known that they define a polytope [45]. That is, a finite number of them dominate all the others. At the time of writing, however, a necessary and sufficient condition for a Boros-Hammer inequality to define a facet of BQP_n is not known.

We remark that switching a Boros-Hammer inequality is remarkably easy. Indeed, to switch on a set $S \subset V_n$, it suffices to change the sign of v_i for all $i \in S$.

Still more valid inequalities for BQP_n can be derived from a connection between BQP_n and the *cut polytope*. This is explained in the next section.

5 Some Related Polytopes

We now review some polytopes that are closely related to the Boolean quadric polytope. Subsection 5.1 deals with the cut polytope, and Subsection 5.2 deals with polytopes that exploit sparsity in the objective function.

5.1 The cut polytope

As before, let $K_n = (V_n, E_n)$ denote the complete graph on n nodes. Given any set $S \subseteq V_n$, we let $\delta(S)$ denote the set of edges in E_n that have exactly one end-node in S. The set $\delta(S)$ is called an *edge-cutset* or simply *cut*. Given an integer $n \geq 3$ and a *weight* $w_e \in \mathbb{Q}$ for all $e \in E_n$, the *max-cut* problem calls for a cut of maximum total weight.

It is well-known (e.g., [6, 14]) that any QUBO instance with n variables can be converted into a max-cut instance with $n + 1$ variables, and vice-versa. This result turns out to have a polyhedral counterpart. Before explaining this, we first present the standard 0-1 LP formulation of the max-cut problem.

For all $e \in E_n$, let z_e be a binary variable, taking the value 1 if and only if e belongs to the cut. The max-cut problem can be formulated as:

$$\max \quad \sum_{e \in E_n} w_e z_e \tag{14}$$

$$\text{s.t. } z_{ij} + z_{ik} + z_{jk} \leq 2 \ \left(\{i, j, k\} \subseteq V_n\right) \tag{15}$$

$$z_{ij} - z_{ik} - z_{jk} \leq 0 \ \left(\{i, j\} \in E_n,\ k \in V_n \setminus \{i, j\}\right) \tag{16}$$

$$\mathbf{z} \in \{0, 1\}^{\binom{n}{2}}. \tag{17}$$

The constraints (15), (16) are (somewhat confusingly) also called *triangle inequalities*.

For a given $n \geq 3$, the convex hull of vectors $\mathbf{z}$ satisfying (15)–(17) is called the *cut polytope* and denoted by CUT_n [7]. To make this clear, consider the case $n = 3$. There are four cut vectors:

$$\begin{pmatrix} z_{12} \\ z_{13} \\ z_{23} \end{pmatrix} \in \left\{ \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \right\}.$$

One can check that these vectors are affinely independent. Thus, CUT_3 is a tetrahedron, as shown in Figure 5.1.

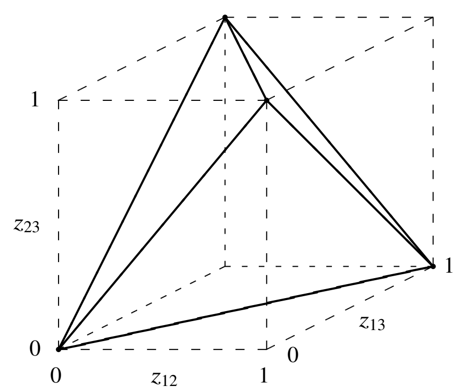

Fig. 4 The cut polytope of order 3.

One can check that the tetrahedron in question is defined by the triangle inequalities $z_{12} + z_{13} + z_{23} \leq 2$, $z_{12} - z_{13} - z_{23} \leq 0$, $z_{13} - z_{13} - z_{23} \leq 0$ and $z_{23} - z_{12} - z_{13} \leq 0$. In other words, for $n = 3$, the triangle inequalities give a complete linear description of CUT_n.

Now recall that BQP_2 was also a tetrahedron. It turns out that BQP_n and CUT_{n+1} are congruent to each other, under a simple (invertible) linear transformation [14, 19, 53]:

Theorem 1 *Let* $\mathbf{x}^* \in \mathbb{R}^n$ *and* $\mathbf{y}^* \in \mathbb{R}^{\binom{n}{2}}$ *be given. Construct a vector* $\mathbf{z}^* \in \mathbb{R}^{\binom{n+1}{2}}$ *as follows:*

$$\begin{aligned} z^*_{i,n+1} &= x^*_i && \big(i \in V_n\big) \\ z^*_{ij} &= x^*_i + x^*_j - 2y^*_{ij} && \big(\{i,j\} \in E_n\big). \end{aligned}$$

Then $(\mathbf{x}^*, \mathbf{y}^*) \in \mathrm{BQP}_n$ *if and only if* $\mathbf{z} \in \mathrm{CUT}_{n+1}$.

The linear transformation in Theorem 1 has come to be known as the *covariance map* [22]. A consequence of Theorem 1 is that the inequality $\alpha^T z \leq \beta$ is valid for CUT_{n+1} if and only if the inequality

$$\sum_{i \in V_n} \left(\sum_{j \in V_{n+1} \setminus \{i\}} \alpha_{ij} \right) x_i - 2 \sum_{e \in E_n} \alpha_e y_e \leq \beta$$

is valid for BQP_n. This enables one to easily convert valid (or facet-defining) inequalities for the cut polytope into valid (or facet-defining) inequalities for the Boolean quadric polytope, and vice-versa.

Example: If we take the inequalities (15) and apply the covariance map, we can obtain the inequalities (5) (if $k = n+1$) or (8) (if $n+1 \notin \{i,j,k\}$). Similarly, if we take the inequalities (16) and apply the covariance map, we can obtain the inequalities (3) and (4) (if we set i or j to $n+1$), the non-negativity inequality $y_{ij} \geq 0$ (if we set k to $n+1$), or the inequality (9) (if $n+1 \notin \{i,j,k\}$). □

Example: If we take the clique inequalities (10) with $|S|$ odd and $s = (|S|-1)/2$, and apply the covariance map, we obtain (with a little work) the following inequalities for the cut polytope:

$$\sum_{\{i,j\} \subset S} z_{ij} \leq \left\lfloor |S|^2/4 \right\rfloor \qquad \big(S \subseteq V_n : |S| \text{ odd}\big). \tag{18}$$

These inequalities were discovered by Barahona & Mahjoub [7]. □

Example: More generally, if we take the Boros-Hammer inequalities (13), and apply the covariance map, we obtain (again with a little work) the following inequalities for the cut polytope:

$$\sum_{\{i,j\} \in E_n} v_i v_j z_{ij} \leq \left\lfloor \frac{\sigma(\mathbf{v})^2}{4} \right\rfloor \qquad \big(\mathbf{v} \in \mathbb{Z}^n : \sigma(\mathbf{v}) \text{ odd}\big). \tag{19}$$

These inequalities were discovered by Deza (see [22]). □

We remark that Laurent & Poljak [44] derived a family of inequalities for CUT_n, called *gap* inequalities, that are even more general than (19). In [24], the gap inequalities are adapted to BQP_n, and then generalised to the case of general mixed-integer quadratic programs.

One can also define a switching operation for the cut polytope [7, 22].

Proposition 2 (Switching for the Cut Polytope) *For an arbitrary set $S \subseteq V_n$, let $\pi^S : \mathbb{R}^{\binom{n}{2}} \mapsto \mathbb{R}^{\binom{n}{2}}$ be the affine transformation that:*

- *replaces z_e with $1 - z_e$ for all $e \in \delta(S)$,*
- *leaves z_e unchanged for all $e \in E_n \setminus \delta(S)$.*

CUT_n *is invariant under this operation.*

This switching operation enables one to take any valid inequality for CUT_n and generate other valid inequalities. For example, if we take the triangle inequality (15) and switch on $\{k\}$, we obtain the triangle inequality (16).

Many other valid inequalities have been discovered for the cut polytope (see [22] for a survey). Among them, we mention only the *odd bicycle wheel* inequalities [7] and the *2-circulant* inequalities [58]. We will see in Section 7 that those particular inequalities are “well-behaved" from an algorithmic viewpoint.

For some other polytopes related to BQP_n see, e.g., [36, 45, 49, 61, 63]. We close this subsection with a remark about the strength of the LP relaxation of the 0-1 LP (14)–(17). Poljak & Tuza [59] showed that, even if all edge-weights are non-negative, the upper bound from the relaxation can be as large as twice the optimum. In other words, the integrality gap can be as large as 100%. For a generalisation of this result, see [5].

5.2 Polytopes which exploit sparsity

A matrix is said to *sparse* if the majority of its elements are zero. Consider a QUBO instance whose quadratic cost matrix $\mathbf{Q}$ is sparse, and assume w.l.o.g. that $\mathbf{Q}$ is symmetric. For all $\{i, j\} \in E_n$ such that $q_{ij} = 0$, we can delete the variable y_{ij} from the formulation (2)–(7), along with the associated constraints. This makes the formulation much smaller and, if we are lucky, much easier to solve. On the other hand, we must take care when deriving valid inequalities: we can no longer use inequalities that involve the variables that have been deleted.

To deal with this from a polyhedral point of view, we need a bit of notation. Let $E = \big\{\{i, j\} \in E_n : q_{ij} \neq 0\big\}$, let $m = |E|$, and let $G = \big(V_n, E\big)$. We define the polytope:

$$\text{BQP}(G) = \text{conv}\Big\{(\mathbf{x}, \mathbf{y}) \in \{0,1\}^{n+m} : \ y_{ij} = x_i x_j \ \big(\{i, j\} \in E\big)\Big\}.$$

Geometrically speaking, $\text{BQP}(G)$ is the *projection* of BQP_n into $\mathbb{R}^{n+m}$.

Unfortunately, projecting a polytope into a subspace is difficult computationally. This makes it harder to derive valid inequalities for BQP(G) than for BQP$_n$. Nevertheless, some useful inequalities are known. In particular, Padberg [53] derived some inequalities called *odd cycle* inequalities, and proved that they define facets of BQP(G). We do not go into details, however, since the notation is rather burdensome.

We can exploit sparsity in the case of the max-cut problem as well. Consider a max-cut instance defined on a graph $G = (V_n, E)$. For a given $S \subseteq V_n$, we let $\delta_G(S)$ denote the set of edges in E that have exactly one end-node in S. We then define the polytope:

$$\text{CUT}(G) = \text{conv}\left\{\mathbf{z} \in \{0,1\}^m : \exists S \subseteq V_n : z_e = 1 \iff e \in \delta_G(S)\right\}.$$

As one might expect, CUT(G) is the projection of CUT$_n$ into $\mathbb{R}^m$.

Barahona & Mahjoub [7] proved the following. Let $\mathcal{C}$ be the set of chordless simple cycles in G. Then a vector $\mathbf{z} \in \{0,1\}^m$ belongs to CUT(G) if and only if it satisfies the following inequalities:

$$\sum_{e \in C \setminus D} z_e \geq \sum_{e \in D} z_e - |D| + 1 \qquad \big(C \in \mathcal{C},\ D \subseteq C : |D| \text{ odd}\big).$$

These inequalities are called *co-circuit* inequalities. Their validity follows from the fact that every cut intersects every cycle an even number of times. Note that the number of co-circuit inequalities can grow exponentially with n. Note also that, when $G = K_n$, every chordless simple cycle is a triangle, and the co-circuit inequalities reduce to the triangle inequalities (15), (16).

It turns out that Padberg's odd cycle inequalities for BQP(G) are precisely the inequalities that can be obtained from the co-circuit inequalities via the covariance map. We omit the proof, for brevity.

6 Some Other Related Convex Sets

In this section, we mention some other important convex sets related to BQP$_n$. Subsection 6.1 presents a non-polyhedral convex set that contains BQP$_n$, and Subsection 6.2 presents the analogous set for CUT$_n$. Then, Subsection 6.3 deals with certain convex cones.

6.1 A non-polyhedral convex set

Our first convex set arises from a certain *semidefinite programming* (SDP) relaxation of QUBO. This set turns out to be non-polyhedral, because an infinite number of linear inequalites are needed to define it. (Informally speaking, it has a “curved" surface.)

The idea of applying SDP to 0-1 quadratic programs is due to Shor [66], and was developed in, e.g., [35, 40, 57]. The basic idea is as follows. We define the $n \times n$ symmetric matrix $\hat{\mathbf{X}} = \mathbf{x}\mathbf{x}^T$, along with the augmented matrix

$$\hat{\mathbf{X}}^+ := \begin{pmatrix} 1 \\ \mathbf{x} \end{pmatrix} \begin{pmatrix} 1 \\ \mathbf{x} \end{pmatrix}^T = \begin{pmatrix} 1 & \mathbf{x}^T \\ \mathbf{x} & \hat{\mathbf{X}} \end{pmatrix}.$$

Since $\hat{\mathbf{X}}^+$ is defined as the product of a vector and its transpose, it should be psd. Moreover, given that $x_i = x_i^2$ for all i, the main diagonal of $\hat{\mathbf{X}}$ should be equal to $\mathbf{x}$. This leads immediately to the following SDP relaxation of QUBO:

$$\max \left\{ \mathbf{Q} \bullet \hat{\mathbf{X}} : \ \operatorname{diag}\left(\hat{\mathbf{X}}\right) = \mathbf{x}, \ \hat{\mathbf{X}}^+ \in \mathcal{S}_+^{n+1} \right\}.$$

To someone who is unfamiliar with nonlinear optimisation, this SDP relaxation may look somewhat mysterious. Fortunately, it can be interpreted in the space of the x and y variables. Indeed, $\hat{\mathbf{X}}^+$ is psd if and only if

$$\begin{pmatrix} s \\ \mathbf{v} \end{pmatrix}^T \begin{pmatrix} 1 & \mathbf{x}^T \\ \mathbf{x} & \hat{\mathbf{X}} \end{pmatrix} \begin{pmatrix} s \\ \mathbf{v} \end{pmatrix} \geq 0 \qquad \left(s \in \mathbb{R}, \ \mathbf{v} \in \mathbb{R}^n\right). \tag{20}$$

Moreover, we have $\hat{x}_{ii} = x_i$ for all $i \in V_n$, and $\hat{x}_{ij} = \hat{x}_{ji} = y_{ij}$ for all $\{i, j\} \in E_n$. Thus, we can write the inequalities (20) in the following form:

$$\sum_{i \in V_n} v_i \left(2s + v_i\right) x_i + 2 \sum_{\{i,j\} \in E_n} v_i v_j y_{ij} + s^2 \geq 0 \qquad \left(s \in \mathbb{R}, \ \mathbf{v} \in \mathbb{R}^n\right). \tag{21}$$

We will call these *psd* inequalities. Note that the psd inequalities are infinite in number.

The psd inequalities include some important inequalities as special cases. For example, if we set v_i to 1, s to 0, and all other components of $\mathbf{v}$ to 0 in (21), we obtain the non-negativity inequality $x_i \geq 0$. If we change s to -1, we obtain $-x_i + 1 \geq 0$, which is equivalent to the upper bound $x_i \leq 1$.

Now consider the following principle submatrix of $\hat{\mathbf{X}}$:

$$\begin{pmatrix} \hat{x}_{ii} & \hat{x}_{ij} \\ \hat{x}_{ij} & \hat{x}_{jj} \end{pmatrix}.$$

Given that the SDP relaxation includes the constraints $x_i = \hat{x}_{ii}$ and $x_j = \hat{x}_{jj}$, this submatrix must be equal to

$$\begin{pmatrix} x_i & \hat{x}_{ij} \\ \hat{x}_{ij} & x_j \end{pmatrix}.$$

Moreover, given that $\hat{\mathbf{X}}$ is psd, this submatrix must have non-negative determinant. Thus, all feasible solutions to the SDP relaxation satisfy $\hat{x}_{ij}^2 \leq x_i x_j$. In other words, the projection into $(\mathbf{x}, \mathbf{y})$-space satisfies $y_{ij}^2 \leq x_i x_j$. This implies in particular that $y_{ij} \leq 1$.

On the other hand, the projection does not satisfy the non-negativity inequalities of the form $y_{ij} \geq 0$. Indeed, one can check that, when $n = 2$, we obtain a feasible solution to the SDP by setting x_1, x_2, $\hat{x}_{11}$ and $\hat{x}_{22}$ to $1/4$, and setting $\hat{x}_{12}$ to $-1/8$. In other words, the SDP relaxation can be strengthened by adding the inequalities $\hat{x}_{ij} \geq 0$ for $1 \leq i < j \leq n$.

6.2 A convex set related to max-cut

As one might expect, the results in the previous subsection have an analogue for the max-cut problem. The SDP relaxation of max-cut was suggested by Schrijver (unpublished) and analysed in, e.g. [29, 43, 56].

We now show how to derive the SDP relaxation. Recall the definitions of $\delta(S)$ and w_e from Subsection 5.1. For each $i \in V_n$, let μ_i be a variable that takes the value 1 if $i \in S$, and -1 otherwise. One can formulate max-cut as the following bivalent quadratic program:

$$\max \left\{ \frac{1}{2} \sum_{\{i,j\} \in E_n} w_{ij} \left(1 - \mu_i \mu_j\right) \, : \, \mu \in \{-1, +1\}^n \right\}.$$

To see that this formulation is valid, note that the quantity $\frac{1}{2}\left(1 - \mu_i \mu_j\right)$ equals 1 if nodes i and j are on opposite shores of the cut, and 0 if they are on the same shore.

Next, we define the matrix $\mathbf{M} = \mu\mu^T$. Note that $\mathbf{M}$ is psd and has 1s on the main diagonal (since $\mu_i^2 = 1$ for all $i \in V_n$). This leads immediately to the SDP relaxation

$$\max \left\{ \frac{1}{2} \sum_{\{i,j\} \in E_n} w_{ij} \left(1 - m_{ij}\right) \, : \, m_{ii} = 1 \left(i \in V_n\right), \, \mathbf{M} \in \mathcal{S}_+^n \right\}.$$

The feasible region of this SDP is called the *elliptope* [43].

Note that the matrix $\mathbf{M}$ is related to the traditional z variables via the identities $m_{ij} = 1 - 2z_{ij}$ for $\{i, j\} \in E_n$. Using this fact, Laurent & Poljak [43] projected the elliptope into $\mathbf{z}$-space. The resulting convex set is defined by the following inequalities:

$$\sum_{\{i,j\} \in E_n} v_i v_j z_{ij} \leq \sigma(\mathbf{v})^2/4 \qquad (\mathbf{v} \in \mathbb{R}^n). \tag{22}$$

One can check that these inequalities are equivalent to the psd inequalities (21), via the covariance map.

It is not hard to see that the inequalities (19) dominate the inequalities (22). This implies in turn that the Boros-Hammer inequalities (13) dominate the psd inequalities (21). See [22] for detailed proofs.

6.3 Cones

There are also several important *convex cones* that are related to BQP_n. To explain them properly, we need to define two more families of inequalities:

- When $\sigma(\mathbf{v}) = 1$, the inequalities (19) reduce to

$$\sum_{\{i,j\} \in E_n} v_i v_j z_{ij} \leq 0 \qquad \big(\mathbf{v} \in \mathbb{Z}^n : \sigma(\mathbf{v}) = 1\big). \tag{23}$$

  These are called *hypermetric* inequalities [17, 18].
- When $\sigma(\mathbf{v}) = 0$, the inequalities (22) reduce to

$$\sum_{\{i,j\} \in E_n} v_i v_j z_{ij} \leq 0 \qquad (\mathbf{v} \in \mathbb{R}^n : \sigma(\mathbf{v}) = 0). \tag{24}$$

  These are called *negative-type* inequalities [19, 64].

Note that the triangle inequalities (16) are hypermetric inequalities. It is also known that the hypermetric inequalities dominate the negative-type inequalities [19].

We now define four convex cones:

- The *cut* cone of order n, which we will call CC_n, is the conic hull of the vectors $\mathbf{z}$ that lie in CUT_n.
- The *metric* cone of order n, denoted by MET_n, is the set of points $\mathbf{z}$ satisfying the triangle inequalities (16).
- The *hypermetric* cone, HYP_n, is the cone defined by the hypermetric inequalities (23).
- The *negative-type* cone, NEG_n, is the cone defined by the negative-type inequalities (24).

From the above considerations, we have $\text{CC}_n \subset \text{HYP}_n \subset \text{MET}_n \cap \text{NEG}_n$. By definition, the cut and metric cones are polyhedral. It has also been shown that the hypermetric cone is polyhedral [21]. The negative-type cone, however, is not. All four cones have interesting applications to the theory of metric spaces and the geometry of numbers; see again [22] for details.

Note that, if we are given a complete linear description of CC_n, we can use switching to get a complete linear description of CUT_n. To see this, let $\bar{\mathbf{z}}$ be an extreme point of CUT_n that is not the origin, and let $\delta(S)$ be the corresponding cut in K_n. If we switch on S, $\bar{\mathbf{z}}$ is mapped to the origin, and any facet containing $\bar{\mathbf{z}}$ is mapped to a facet containing the origin. Reversing this argument, we can obtain any inequality that defines a facet of CUT_n by switching an inequality that defines a facet of CC_n.

Using the covariance map, one can derive analogous inequalities and cones in $(\mathbf{x}, \mathbf{y})$-space. For the sake of brevity, we do not go into details. We just mention that the hypermetric inequalities (23) map to the following inequalities for BQP_n:

$$\sum_{i \in V_n} v_i(1 - v_i)x_i \le 2 \sum_{1 \le i < j \le n} v_i v_j y_{ij} \qquad (\mathbf{v} \in \mathbb{Z}^n). \tag{25}$$

These are called *hypermetric correlation* inequalities [20]. We remark that they can be viewed as the special case of the Boros-Hammer inequalities (13) in which $s = 0$.

7 Algorithms

Now we turn to the algorithmic implications of the above results. Subsection 7.1 describes a (fairly) simple exact algorithm, called *cut-and-branch*. Subsection 7.2 concerns subroutines that search for useful cutting planes. Finally, Subsection 7.3 describes a more sophisticated algorithmic framework, called *branch-and-cut*.

7.1 Cut-and-branch

Suppose we wish to solve an ILP of the form (1). The continuous relaxation of the ILP is an LP, which can be solved with, e.g., the simplex method. This yields a solution, say $\mathbf{x}^*$. If $\mathbf{x}^*$ is integral, we have solved the ILP. If not, the quantity $\mathbf{c}^T\mathbf{x}^*$ gives an *upper bound* on the optimal profit.

At this point, we can resort to an old-fashioned solution method, such as Gomory's cutting-plane method [30] or branch-and-bound [42]. A more effective approach, first suggested by Crowder *et al.* [13], is to add some *strong* (preferably facet-defining) valid inequalities to the formulation, and then invoke branch-and-bound. The resulting algorithm, which has come to be known as "cut-and-branch", is outlined in Algorithm 1.

The idea behind cut-and-branch is that the cutting planes typically yield a significant decrease in the upper bound. This in turn leads to a reduction in the size of the branch-and-bound tree. The results in [13] indicate that, for many ILPs arising in practice, the reduction in the size of the tree can be dramatic, and enable one to solve ILPs that are unsolvable with traditional branch-and-bound.

Cut-and-branch algorithms for QUBO and related problems can be found in, e.g., [6, 23, 35, 49].

7.2 Separation algorithms

In Algorithm 1, there is a line that says "Search for strong valid inequalities that are violated by $\mathbf{x}^*$". Geometrically speaking, finding such inequalities (cutting planes) amounts to finding a hyperplane that "separates" the current fractional LP solution

Algorithm 1: Cut-and-Branch Algorithm

```
input  : positive integers n and m; matrix A; vectors b, c
Solve the LP relaxation and let x* be the solution;
repeat
    if x* is integer then
        Output x* and quit;
    end
    Search for strong valid inequalities that are violated by x*;
    if at least one inequality has been found then
        Add one or more inequalities to the LP as cutting planes;
        Re-optimise the LP and update x*;
    end
until no more violated inequalities are found;
Optional: Delete all cutting planes that have a positive slack;
Declare all variables integer;
Feed the resulting ILP into a branch-and-bound solver;
Let x* be the solution;
output  Optimal solution x*
:
```

from the feasible integer solutions. For this reason, algorithms that search for cutting planes are called “separation algorithms" [32]. More precisely:

- An *exact separation algorithm* for a given family of valid inequalities is an algorithm that takes a fractional LP solution as input, and outputs one or more violated inequalities in the given family, if any exist.
- A *heuristic separation algorithm* for a given family of valid inequalities is an algorithm that takes a fractional LP solution as input, and outputs either one or more violated inequalities in the given family, or a failure message.

In the context of QUBO, we can assume that the fractional solution is a pair $\left(\mathbf{x}^*, \mathbf{y}^*\right) \in [0,1]^{n+\binom{n}{2}}$.

The separation problem for the triangle inequalities (8), (9) can be solved in $O\left(n^3\right)$ time by brute-force enumeration. The complexity of the separation problems for the inequalities (10)–(13) is unknown, but we suspect that they are all $\mathcal{NP}$-hard. Greedy separation heuristics for the inequalities (10)–(12) can be found in, e.g., [48, 67, 69].

The separation problem for the psd inequalities (21) can be solved in polynomial time [32]. The following method works well in practice (e.g., [34, 65]). Given $\left(\mathbf{x}^*, \mathbf{y}^*\right)$, construct the matrix $\hat{\mathbf{X}}^{+}$. Find the minimum eigenvalue of $\hat{\mathbf{X}}^{+}$, to some desired precision. If the eigenvalue is non-negative, stop. Otherwise, find the associated eigenvector, again to the desired precision. Write the eigenvector as $\binom{s^*}{\mathbf{v}^*}$. This eigenvector yields a violated psd inequality. To see why, let $\lambda < 0$ be the eigenvalue, and note that

$$\begin{pmatrix} s^* \\ \mathbf{v}^* \end{pmatrix}^T \begin{pmatrix} 1 & \mathbf{x}^T \\ \mathbf{x} & \hat{\mathbf{X}} \end{pmatrix} \begin{pmatrix} s^* \\ \mathbf{v}^* \end{pmatrix} = \begin{pmatrix} s^* \\ \mathbf{v}^* \end{pmatrix}^T \left(\lambda \begin{pmatrix} s^* \\ \mathbf{v}^* \end{pmatrix} \right) = \lambda \left\| \begin{pmatrix} s^* \\ \mathbf{v}^* \end{pmatrix} \right\|_2^2 < 0.$$

Several polynomial-time separation algorithms are known for the cut polytope. The separation problem for the triangle inequalities (15), (16) can be solved in $O(n^3)$ time by enumeration. Gerards [27] presented an $O(n^5)$ separation algorithm for the odd bicycle wheel inequalities. There are also $O(n^5)$ separation algorithms for various generalisations of the 2-circulant inequalities [37,38,45]. The separation problems for the inequalities (22) and (24) can be solved in a similar way to the psd inequalities (see [22]).

At the time of writing, the complexity of separation is unknown for the remaining inequalities for the cut polytope. In [35], a greedy separation heuristic is presented for the inequalities (18) and their switchings. Separation heuristics for the inequalities (19) can be found in [16, 25, 34, 35]. A separation heuristic for the gap inequalities is given in [25].

Letchford & Sørensen [46] showed that the separation problems for the inequalities (13), (19), (23) and (25) are equivalent. That is, either all of them can be solved in polynomial time, or none of them can. See also Avis [4].

Finally, we mention that there are several exact and heuristic separation algorithms designed for *sparse* QUBO and max-cut instances (e.g., [6–8, 11, 47]). We omit details, for brevity.

7.3 Branch-and-cut

Now that we have explained the concept of separation, we return to solution algorithms. Recall that, in cut-and-branch, we are permitted to add cutting planes only *before* running branch-and-bound. A natural extension is to permit the addition of cutting planes *while* running branch-and-bound.

To make this more precise, we recall that branch-and-bound is a recursive algorithm, which solves a series of LP subproblems, arranged in a tree structure. Suppose that we have just solved the LP that corresponds to one particular branch of the tree. If the solution is fractional, we can run one or more separation algorithms, in an attempt to cut it off. If any cutting planes are found, we can add them to the LP, re-optimise, and repeat. This causes the upper bound at that branch to decrease, which may allow one to eliminate the branch from consideration.

This approach was discovered by several authors, apparently independently (e.g., [31, 50, 54]). It was given the name *branch-and-cut* by Padberg and Rinaldi [54]. It works remarkably well, and has been applied to a wide range of problems in integer programming and combinatorial optimisation [10, 51].

Although branch-and-cut is conceptually simple, it requires considerably more programming effort than cut-and-branch. The main reason is that the branch-and-bound solver can no longer be treated as a “black box". Moreover, some additiona implementation “tricks" are needed to make the approach work efficiently. Several such tricks are given in [54], such as (a) starting with a subset of the variables and generating the others dynamically, (b) storing cutting planes in a “cut pool", (c) scanning the cut pool before calling the more time-consuming separation routines,

(d) deleting non-binding cutting planes to save memory, and (e) producing heuristic integer solutions by rounding fractional values to integers.

Oddly, no-one has yet designed and implemented a full branch-and-cut algorithm for QUBO. Indeed, at present, the most effective exact algorithms for QUBO use SDP relaxations, with triangle inequalities incorporated via Lagrangian relaxation (see, e.g., [41, 62]).

8 Concluding Remarks

The Boolean quadric and cut polytopes have been studied in depth, and many families of strong valid linear inequalities are now known. For some of the families, we also have efficient exact or heuristic separation algorithms.

There remain several interesting directions for possible future research. Among them, we mention the following:

- Determine whether or not the separation problem for the hypermetric inequalities (23) can be solved in polynomial time.
- Design, implement and test a full branch-and-cut algorithm for QUBO and related problems.
- Understand better the relative advantages and disadvantages of LP-based and SDP-based approaches to QUBO.
- Provide an open source library of separation algorithms for QUBO and related problems.

References

1. K. Aardal & C.P.M. Van Hoesel (1996) Polyhedral techniques in combinatorial optimization I: theory. *Statistica Neerlandica*, 50, 3–26.
2. K. Aardal & C.P.M. Van Hoesel (1999) Polyhedral techniques in combinatorial optimization II: applications and computations. *Statistica Neerlandica*, 53, 131–177.
3. K. Aardal & R. Weismantel (1997) Polyhedral combinatorics. In M. Dell'Amico, F. Maffioli & S. Martello (eds) *Annotated Bibliographies in Combnatorial Optimization*, pp. 31–44. New York: Wiley.
4. D. Avis (2003) On the complexity of testing hypermetric, negative type, k-gonal and gap inequalities. In J. Akiyama & M. Kanö (eds) *Discrete and Computational Geometry*, pp. 51–59. Berlin: Springer.
5. D. Avis & J. Umemoto (2003) Stronger linear programming relaxations of max-cut. *Math. Program.*, 97, 451–469.
6. F. Barahona, M. Jünger & G. Reinelt (1989) Experiments in quadratic 0-1 programming. *Math. Program.*, 44, 127–137.
7. F. Barahona & A.R. Mahjoub (1986) On the cut polytope. *Math. Program.*, 36, 157–173.
8. T. Bonato, M. Jünger, G. Reinelt & G. Rinaldi (2014) Lifting and separation procedures for the cut polytope. *Math. Program.*, 146, 351–378.
9. E. Boros & P.L. Hammer (1993) Cut-polytopes, Boolean quadric polytopes and nonnegative quadratic pseudo-Boolean functions. *Math. Oper. Res.*, 18, 245–253.

10. A. Caprara & M. Fischetti (1997) Branch-and-cut algorithms. In M. Dell'Amico, F. Maffioli & S. Martello (eds.) *Annotated Bibliographies in Combinatorial Optimization*, pp. 45–64. New York: Wiley.
11. E. Cheng (1998) Separating subdivision of bicycle wheel inequalities over cut polytopes. *Oper. Res. Lett.*, 23, 13–19.
12. W. Cook (2010) Fifty-plus years of combinatorial integer programming. In M. Juenger *et al.* (eds.) *50 Years of Integer Programming*, pp. 387–430. Heidelberg: Springer.
13. H. Crowder, E.L. Johnson & M. Padberg (1983) Solving large-scale zero-one linear programming problems. *Oper. Res.*, 31, 803–834.
14. C. De Simone (1989) The cut polytope and the Boolean quadric polytope. *Discr. Math.*, 79, 71–75.
15. C. De Simone (1996) A note on the Boolean quadric polytope. *Oper. Res. Lett.*, 19, 115–116.
16. C. De Simone & G. Rinaldi (1994) A cutting plane algorithm for the max-cut problem. *Optim. Methods Softw.*, 3, 195–214.
17. M. Deza (1961) On the Hamming geometry of unitary cubes. *Soviet Physics Doklady*, 5, 940–943.
18. M. Deza (1962) Realizability of distance matrices in unit cubes (in Russian). *Problemy Kibernitiki*, 7, 31–42.
19. M. Deza (1973) Matrices de formes quadratiques non négatives pour des arguments binaires. *Comptes rendus de l'Académie des Sciences de Paris*, 277, 873–875.
20. M. Deza & V.P. Grishukhin (1998) Voronoi L-decomposition of PSD_n and the hypermetric correlation cone. In: P. Engel & H. Syta (eds) *Voronoi's Impact on Modern Science*. Kiev, Ukraine: Institute of Mathematics of the National Academy of Science.
21. M. Deza, V.P. Grishukhin & M. Laurent (1993) The hypermetric cone is polyhedral. *Combinatorica*, 13, 1–15.
22. M. Deza & M. Laurent (1997) *Geometry of Cuts and Metrics*. Berlin: Springer.
23. F. Djeumou Fomeni, K. Kaparis & A.N. Letchford (2020) A cut-and-branch algorithm for the quadratic knapsack problem. *Discr. Optim.*, to appear.
24. L. Galli, K. Kaparis & A.N. Letchford (2011) Gap inequalities for non-convex mixed-integer quadratic programs. *Oper. Res. Lett.*, 39, 297–300.
25. L. Galli, K. Kaparis & A.N. Letchford (2012) Gap inequalities for the max-cut problem: a cutting-plane algorithm. In A.R Mahjoub *et al.* (eds) *Combinatorial Optimization: 2nd International Symposium*, pp. 178–188. Berlin: Springer.
26. M.R. Garey, D.S. Johnson & L.J. Stockmeyer (1976) Some simplified $\mathcal{NP}$-complete graph problems. *Theor. Comput. Sci.*, 1, 237–267.
27. A.M.H. Gerards (1985) Testing the odd bicycle wheel inequalities for the bipartite subgraph polytope. *Math. Oper. Res.*, 10, 359–360.
28. F. Glover & E. Woolsey (1974) Converting the 0-1 polynomial program to a 0-1 linear program. *Oper. Res.*, 22, 180–182.
29. M.X. Goemans & D.P. Williamson (1995) Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. Ass. Comp. Mach.* 42, 1115–1145.
30. R.E. Gomory (1958) Outline of an algorithm for integer solutions to linear programs. *Bull. Amer. Math. Soc.*, 64, 275–278.
31. M. Grötschel, M. Jünger & G. Reinelt (1984) A cutting plane algorithm for the linear ordering problem. *Oper. Res.*, 32, 1195–1220.
32. M. Grötschel, L. Lovász & A. Schijver (1988) *Geometric Algorithms and Combinatorial Optimization* (2nd edn). Heidelberg: Springer.
33. M. Grötschel & M.W. Padberg (1985) Polyhedral theory. In E.L. Lawler *et al.* (eds) *The Travelling Salesman Problem: A Guided Tour of Combinatorial Optimization*. New York: Wiley.
34. G. Gruber (2000) *On Semidefinite Programming and Applications in Combinatorial Optimization*. PhD thesis, Department of Mathematics, University of Klagenfurt.
35. C. Helmberg & F. Rendl (1998) Solving quadratic (0,1)-programs by semidefinite programs and cutting planes. *Math. Program.*, 82, 291–315.

36. M. Jünger & V. Kaibel (2001) Box-inequalities for quadratic assignment polytopes. *Math. Program.*, 91, 175–197.
37. K. Kaparis & A.N. Letchford (2018) A note on the 2-circulant inequalities for the max-cut problem. *Oper. Res. Lett.*, 46, 443–447.
38. K. Kaparis, A.N. Letchford & Y. Mourtos (2022) Generalised 2-circulant inequalities for the max-cut problem. *Oper. Res. Lett.*, 50, 122–128.
39. R.M. Karp & C.H. Papadimitriou (1982) On linear characterizations of combinatorial optimization problems. *SIAM J. Comput.*, 11, 620–632.
40. F. Koerner (1988) A tight bound for the Boolean quadratic optimization problem and its use in a branch and bound algorithm. *Optimization*, 19, 711–721.
41. N. Krislock, J. Malick & F. Roupin (2014) Improved semidefinite bounding procedure for solving max-cut problems to optimality. *Math. Program.*, 143, 61–86.
42. A.H. Land & A.G. Doig (1960) An automatic method of solving discrete programming problems. *Econometrica*, 28, 497–520.
43. M. Laurent & S. Poljak (1995) On a positive semidefinite relaxation of the cut polytope. *Lin. Alg. & Appl.*, 223-4, 439–461.
44. M. Laurent & S. Poljak (1996) Gap inequalities for the cut polytope. *Eur. J. Combinatorics*, 17, 233–254.
45. A.N. Letchford & M.M. Sørensen (2012) Binary positive semidefinite matrices and associated integer polytopes. *Math. Program.*, 131, 253–271.
46. A.N. Letchford & M.M. Sørensen (2014) A new separation algorithm for the Boolean quadric and cut polytopes. *Discr. Optim.*, 14, 61–71.
47. F. Liers (2004) *Contributions to Determining Exact Ground-States of Ising Spin-Glasses and to their Physics*. PhD thesis, Department of Mathematics and Computer Science, University of Cologne.
48. E.M. Macambira & C.C. de Souza (2000) The edge-weighted clique problem: valid inequalities, facets and polyhedral computations. *Eur. J. Oper. Res.*, 123, 346–371.
49. A. Mehrotra (1997) Cardinality constrained Boolean quadratic polytope. *Discr. Appl. Math.*, 79, 137–154.
50. P. Miliotis (1976) Integer programming approaches to the travelling salesman problem. *Math. Program.*, 10, 367-378.
51. J.E. Mitchell (2010) Branch and cut. In J.J. Cochran *et al.* (eds) *Encyclopedia of Operations Research and Management Science*. New York: Wiley.
52. G.L. Nemhauser & L.A. Wolsey (1988) *Integer and Combinatorial Optimization*. New York: Wiley.
53. M.W. Padberg (1989) The Boolean quadric polytope: some characteristics, facets and relatives. *Math. Program.*, 45, 139–172.
54. M.W. Padberg & G. Rinaldi (1991) A branch-and-cut algorithm for the resolution of large-scale symmetric travelling salesman problems. *SIAM Rev.*, 33, 60–100.
55. I. Pitowsky (1991) Correlation polytopes: their geometry and complexity. *Math. Program.*, 50, 395–414.
56. S. Poljak & F. Rendl (1995) Non-polyhedral relaxations of graph-bisection problems. *SIAM J. Optim.*, 5, 467–487.
57. S. Poljak, F. Rendl & H. Wolkowicz (1995) A recipe for semidefinite relaxation for (0,1)-quadratic programming. *J. Glob. Optim.*, 7, 51–73.
58. S. Poljak & D. Turzik (1992) Max-cut in circulant graphs. *Discr. Math.*, 108, 379–392.
59. S. Poljak & Zs. Tuza (1994) The expected relative error of the polyhedral approximation of the max-cut problem. *Oper. Res. Lett.*, 16, 191–198.
60. A.K. Pujari, A.K. Mittal & S.K. Gupta (1983) A convex polytope of diameter one. *Discr. Appl. Math.*, 5, 241–242.
61. D.J. Rader (1997) Valid inequalities and facets of the quadratic 0-1 knapsack polytope. *RUTCOR Research Report 16-97*, Rutgers University.
62. F. Rendl, G. Rinaldi & A. Wiegele (2010) Solving max-cut to optimality by intersecting semidefinite and polyhedral relaxations. *Math. Program.*, 121, 307–355.

63. H. Saito, T. Fujie, T. Matsui & S. Matuura (2009) A study of the quadratic semi-assignment polytope. *Discr. Optim.*, 6, 37–50.
64. I.J. Schoenberg (1938) Metric spaces and positive definite functions. *Trans. Amer. Math. Soc.*, 44, 522–536.
65. H.D. Sherali & B.M.P. Fraticelli (2002) Enhancing RLT relaxations via a new class of semidefinite cuts. *J. Glob. Optim.*, 22, 233–261.
66. N.Z. Shor (1987) Quadratic optimization problems. *Tekhnicheskaya Kibernetika*, 1, 128–139.
67. M.M. Sørensen (2004) New facets and a branch-and-cut algorithm for the weighted clique problem. *Eur. J. Oper. Res.*, 154, 57–70.
68. H. Weyl (1935) Elementare Theorie der konvexen Polyeder. *Commentarii Math. Helvetici*, 7, 290–306.
69. Y. Yajima & T. Fujie (1998) A polyhedral approach for nonconvex quadratic programming problems with box constraints. *J. Glob. Optim.*, 13, 151–170.