Operations Research Letters 46 (2018) 443–447

Contents lists available at ScienceDirect

Operations Research Letters

journal homepage: www.elsevier.com/locate/orl

A note on the 2-circulant inequalities for the max-cut problem

Konstantinos Kaparis [a,*], Adam N. Letchford [b]

[a] Department of Business Administration, University of Macedonia, Greece
[b] Department of Management Science, Lancaster University, United Kingdom

ARTICLE INFO

Article history:
Received 12 October 2017
Received in revised form 1 February 2018
Accepted 23 May 2018
Available online 30 May 2018

Keywords:
Max-cut problem
Polyhedral combinatorics
Branch-and-cut

ABSTRACT

The *max-cut* problem is a much-studied $\mathcal{NP}$-hard combinatorial optimisation problem. Poljak and Turzik found some facet-defining inequalities for this problem, which we call *2-circulant* inequalities. Two polynomial-time separation algorithms have been found for these inequalities, but one is very slow and the other is very complicated. We present a third algorithm, which is as fast as the faster of the existing two, but much simpler.

1. Introduction

Let G be an undirected graph, with vertex set V and edge set E. For any $S \subseteq V$, the set of edges having exactly one end-vertex in S is called an *edge-cutset* or simply *cut*. Given a weight $w_e \in \mathbb{R}$ for each edge $e \in E$, the *max-cut* problem calls for a cut in G of maximum total weight.

The max-cut problem is strongly $\mathcal{NP}$-hard [13]. It has many applications and has received much attention (see, e.g., [10,18]). As present, the most successful exact algorithms for max-cut are based on either *linear programming* (LP) or *semidefinite programming* (SDP) relaxations (see, e.g., [4,23,25]). LP-based algorithms can solve sparse instances with up to around 500 nodes, and SDP-based ones can solve dense instances with up to around 150 nodes.

To construct strong LP relaxations of combinatorial optimisation problems, it helps to derive families of strong (preferably facet-defining) valid linear inequalities (see, e.g., [7]). Several such families have been derived for max-cut (see, e.g., [3,10]). In this paper, we focus on some inequalities discovered by Poljak & Turzik [24], which we call 2-*circulant* inequalities.

The *separation problem* for a given family of valid linear inequalities is this: given a fractional solution to an LP relaxation of an instance of the max-cut problem, either find an inequality in the family that is violated by the fractional solution, or prove that no such violated inequality exists (see, e.g., [7,16]). In [24], Poljak and Turzik claimed (without proof) that the separation problem for the 2-circulant inequalities (or more precisely a generalisation of them) could be solved in polynomial time. Explicit algorithms for this purpose were however discovered only later on [20,21].

Unfortunately, the separation algorithm in [20] is very slow, involving the solution of $O(n^2)$ very large LPs (where $n = |V|$). The algorithm in [21] is faster, with a running time of $O(n^5)$, but it is very hard to implement. The purpose of this paper is to present a third algorithm, which is much simpler than both of the other algorithms, yet still runs in $O(n^5)$ time.

The paper has a simple structure: the literature is reviewed in Section 2, the new algorithms are presented in Section 3, and concluding remarks are made in Section 5. Throughout the paper, we let K_n denote the complete undirected graph with vertex set $V_n = \{1, \ldots, n\}$ and edge set $E_n = \{e \subset V_n : |e| = 2\}$. 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. We also assume that $n \geq 5$ throughout the paper, to avoid trivial cases.

2. Literature review

Now we review the literature. Due to space limitations, we only review essential results, and refer the reader to [10,18] for comprehensive surveys.

2.1. The cut polytope

By adding dummy edges of weight zero, if necessary, we can work with the complete graph K_n. The max-cut problem can then be formulated as the following 0-1 LP:

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

* Corresponding author.
E-mail addresses: k.kaparis@uom.edu.gr (K. Kaparis), A.N.Letchford@lancaster.ac.uk (A.N. Letchford).

https://doi.org/10.1016/j.orl.2018.05.006
0167-6377/

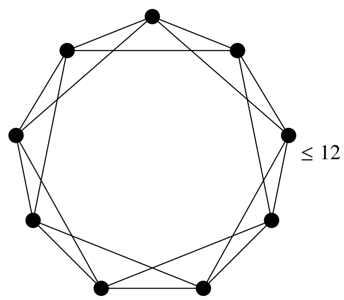

Fig. 1. Representation of a 2-circulant inequality with $c = 9$.

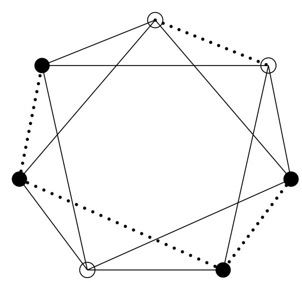

Fig. 2. A 2-circulant with $c = 7$ contains a cut with $(3c-1)/2 = 10$ edges. Nodes in S represented by filled circles, others by hollow circles. Edges in $\delta(S)$ represented by solid lines, others by dotted lines.

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.

Barahona and Mahjoub [3] call the convex hull of feasible solutions to the above 0-1 LP the *cut polytope*. We will denote it by CUT_n. They show that the triangle inequalities define facets of CUT_n, along with various other families of inequalities, including so-called *odd clique* and *odd bicycle wheel* inequalities. Since then, a huge array of valid and facet-defining inequalities has been discovered (see Part V of [10]). We will be particularly interested in the following inequalities, due to Poljak & Turzik [24]:

Proposition 1 (*Poljak & Turzik, 1992 [24]*). *Let c be a positive integer with $5 \le c \le n$, and suppose that c is congruent to 1 modulo 4. Let $v_1, \ldots, v_c$ be distinct vertices in K_n. The inequality*

$$\sum_{i=1}^{c} \Big(x\big(v_i, v_{i+1}\big) + x\big(v_i, v_{i+2}\big) \Big) \le 3(c-1)/2 \tag{3}$$

defines a facet of CUT_n, where indices are taken modulo c.

We will call the inequalities (3) 2-*circulant* inequalities. Fig. 1 gives a graphical representation of a 2-circulant inequality with $c = 9$. Small filled circles represent nodes, and edges represent variables with a coefficient of one on the left-hand side.

A point that will be important later on (see Section 2.3) is that 2-circulant inequalities are not valid for CUT_n when c is congruent to 3 mod 4. Indeed, to make them valid in that case, it is necessary to increase the right-hand side by one, to $(3c-1)/2$ (see Fig. 2). Moreover, the resulting inequalities are not of interest, since they are easily shown to be implied by triangle inequalities.

We remark that the cut polytope is closely related to the so-called *Boolean quadric polytope*, which is a polytope associated with unconstrained quadratic programming in binary variables (see, e.g., [2,8,10,22]). For brevity, we do not go into details, but simply point out that our new separation algorithms can be easily adapted to the Boolean quadric polytope.

2.2. Switching

An important operation, called *switching*, was also defined in [3]. It states that, if the inequality $\lambda^T x \le \gamma$ is valid (or facet-defining) for CUT_n, then the 'switched' inequality

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

is also valid (or facet-defining), for any $S \subset V_n$. As a simple example, taking a triangle inequality of the form (1) and switching on node k, we obtain a triangle inequality of the form (2). In a similar way, one can construct switched odd clique, odd bicycle wheel and 2-circulant inequalities.

2.3. Separation

Known results on exact separation algorithms include the following:

- Separation for the triangle inequalities (1), (2) can be solved in $O(n^3)$ time by mere enumeration.
- More generally, separation for (switched) odd clique inequalities on k nodes, for a fixed odd $k \ge 3$, can be solved by enumeration in $O(n^k)$ time.
- Separation for the odd bicycle wheel inequalities can be solved in $O(n^5)$ time [14]. The idea is to reduce the separation problem to $\binom{n}{2}$ minimum-weight odd cycle problems in K_{n-2}, and then use the known fact [3,15] that the minimum-weight odd cycle problem can itself be reduced to a series of shortest (s,t)-path problems.
- The separation problem for a family of valid inequalities that arises from an SDP relaxation of max-cut can be solved (to arbitrary fixed precision) in polynomial time, by computing the minimum Eigenvalue of a certain matrix [19]. The inequalities in question are however never facet-defining.
- One can separate in polynomial time over families of valid inequalities that include not only all odd bicycle wheel and 2-circulant inequalities, but also all of their switchings [20,21]. The algorithm in [20] is based on lift-and-project (see [1]), and involves the solution of $\binom{n}{2}$ LPs, each with $O(n^3)$ variables and $O(n^2)$ constraints. The algorithm in [21] is based on repeated applications of the covariance map, along with arguments based on "$\{0, \frac{1}{2}\}$-cuts" (see [5]). It runs in $O(n^5)$ time, but is very hard both to understand and to implement.

Poljak & Turzik [24] noted (Remark 5.4, page 391) that 2-circulant inequalities remain valid even when the nodes $v_1, \ldots, v_c$ are not all distinct, provided that $v_i \ne v_{i+1}$ for $i = 1, \ldots, c$. They also stated that the separation problem for the resulting generalised 2-circulant inequalities could be transformed to a minimum-weight odd cycle problem on a suitable graph, but they did not give a proof. (Actually, we believe that such a transformation is impossible, due to the fact that 2-circulant inequalities are valid not when c is odd, but only when c is congruent to 1 mod 4.)

There are also several separation *heuristics* available for CUT_n and related polyhedra; see, e.g., [2,4,6,9,12,17]. For the sake of brevity, we do not go into details.

3. The new separation algorithms

In this section, we describe our new separation algorithms. Section 3.1 deals with the 2-circulant inequalities themselves, whereas Section 3.2 deals with switched 2-circulant inequalities. The inputs to both algorithms are an integer $n \ge 5$ and a fractional point $x^* \in [0,1]^{E_n}$ that lies outside CUT_n.