Facets from Gadgets

Adam N. Letchford[*] Anh N. Vu[†]

To appear in *Mathematical Programming Series A*

Abstract

We present a new tool for generating cutting planes for $\mathcal{NP}$-hard combinatorial optimisation problems. It is based on the concept of *gadgets* — small subproblems that are "glued" together to form hard problems — which we borrow from the literature on computational complexity. Using gadgets, we are able to derive huge (exponentially large) new families of strong (and sometimes facet-defining) cutting planes, accompanied by efficient separation algorithms. We illustrate the power of this approach on the *asymmetric traveling salesman*, *stable set* and *clique partitioning* problems.[1]

Keywords: branch-and-cut; gadgets; traveling salesman problem; stable set problem; clique partitioning problem

1 Introduction

A popular way to solve $\mathcal{NP}$-hard *combinatorial optimisation problems* (COPs) to proven (near-)optimality is *branch-and-cut* [37]. In this approach, the COP is formulated as an *integer linear program* (ILP), and the convex hull of feasible solutions is studied, with the aim of deriving families of strong valid linear inequalities (called *cutting planes* or simply *cuts*). These cuts are then used to construct strong linear programming relaxations of the ILP. Algorithms for generating cuts are called *separation algorithms*, and the strongest possible cuts are ones that define *facets* of the convex hull (see, e.g., [13, 26]).

The purpose of this paper is to present a new tool for generating cuts. It is based on the concept of *gadgets* — small subproblems that are "glued" together to form hard problems — which we borrow from the literature on computational complexity theory (see, e.g., [19, 38]). Using gadgets, we

[*]Corresponding author. Department of Management Science, Lancaster University, Lancaster LA1 4YX, United Kingdom. E-mail: `a.n.letchford@lancaster.ac.uk`

[†]Former PhD student at Lancaster. E-mail: `a.vu32@yahoo.com`

[1]This paper is dedicated to the memory of Alberto Caprara (1968–2012).

are able to derive huge (exponentially large) new families of cuts. We also obtain efficient separation algorithms as a by-product of the analysis.

We illustrate the power of the gadget approach on three well-known $\mathcal{NP}$-hard COPs: the *asymmetric traveling salesman problem* (ATSP), the *stable set problem* (SSP) and the *clique partitioning problem* (CPP). For all three problems, some of the cuts that we derive are facet-defining. Interestingly, some of the cuts that we derive for the CPP also have *rank* greater than one, in the sense of Chvátal [12].

The structure of the paper is as follows. In Section 2, we review the relevant literature. In Section 3, we define gadgets and show how they can be used both to derive cuts and solve the associated separation problem. In Section 4, we apply the gadget approach to the three COPs mentioned above. Finally, in Section 5, we make some concluding remarks.

Throughout the paper, we use the following (standard) graph-theoretic notation and terminology. $G = (V, E)$ denotes a generic (simple, loopless) undirected graph with vertex set V and edge set E. Given a node $i \in V$, we let $n(i)$ denote the *neighbours* of i in G, i.e., the set of nodes adjacent to i. A *clique* (respectively, *stable set*) is a set of pairwise adjacent (respectively, non-adjacent) nodes. A *cycle* is a connected subgraph in which all nodes have degree two. Given a cycle C, we let $V(C)$ denote its vertex set and $E(C)$ its edge set. If some of the edges of G are labelled "odd", then a cycle containing an odd number of odd edges is itself called "odd". Given a positive integer n, we let $K_n = (V_n, E_n)$ denote the complete graph on n nodes, where $V_n = \{1, \ldots, n\}$ and $E_n = \{e \subset V_n : |e| = 2\}$. Finally, for directed graphs (digraphs), we write $G = (V, A)$, where A is the set of *arcs*. We also let $D_n = (V_n, A_n)$ denote the complete digraph on n nodes, where $A_n = \{(i, j) : i \in V_n, j \in V_n \setminus \{i\}\}$.

2 Literature Review

We now review the relevant literature. In Subsection 2.1, we review an existing scheme for generating cuts, which will turn out to be a special case of our scheme. For reasons which will become apparent, we describe the *satisfiability* problem in Subsection 2.2. We cover *gadgets* in Subsection 2.3. Finally, in Subsection 2.4, we recall some basic facts about the ATSP, SSP and CPP.

2.1 $\{0, \frac{1}{2}\}$-cuts and odd cycles

Our new scheme for generating cuts was inspired by an existing scheme, due to Caprara & Fischetti [6]. Let $S \subset \mathbb{Z}^n$ be the set of solutions to an ILP, let P be the convex hull of S, and let $Ax \leq b$ be an arbitrary collection of valid linear inequalities for P, with $b \in \mathbb{Z}^m$ and $A \in \mathbb{Z}^{mn}$. Caprara and

Fischetti call a cutting plane a "$\{0, \frac{1}{2}\}$-cut" (w.r.t. the given collection of inequalities) if it can be written in the form

$$\left(\lambda^T A\right) x \;\leq\; \left\lfloor \lambda^T b \right\rfloor,$$

for some $\lambda \in \{0, \frac{1}{2}\}^m$ such that $\lambda^T A \in \mathbb{Z}^n$ and $\lambda^T b \notin \mathbb{Z}$.

Caprara and Fischetti show that the separation problem for $\{0, \frac{1}{2}\}$-cuts can be solved in polynomial time under certain conditions. In particular, if every row of A has exactly two odd coefficients, then separation is equivalent to finding a minimum-weight odd cycle in a graph with n nodes and m edges. This problem in turn can be reduced to a series of n shortest-path problems [4, 20]. If we use an efficient implementation of Dijkstra's method to solve these shortest-path problems (see [18]), we can compute the minimum-weight odd cycle in $O\left(nm + n^2 \log n\right)$ time.

In the remainder of the paper, when we refer to "the Caprara–Fischetti approach", we mean the one based on odd cycles. (They also describe another approach, based on odd cuts.) Some related schemes for generating cuts can be found in, e.g., [5, 27, 32].

2.2 Satisfiability

The *satisfiability* problem (`SAT`) was the first ever problem to be proved $\mathcal{NP}$-complete [14]. In this problem, we have a collection of Boolean variables, say $x_1, \ldots, x_n$, and a collection of logical disjunctions, called *clauses*, that they should satisfy. An example of a clause is $x_1 \vee \bar{x}_2 \vee \bar{x}_3$, which should be read as "x_1 is true or x_2 is false or x_3 is false". In this context, "or" is always intended to be inclusive. The task is to check whether there exists an assignment of truth values to the variables that satisfies all of the clauses.

The special case of `SAT` in which every clause contains exactly k terms is called `k-SAT`. It is well known that `2-SAT` can be solved in linear time. One way to do it as as follows [1]. Replace each clause with two *implications*. For example, the clause $x_1 \vee \bar{x}_2$ is replaced with the statements "$\bar{x}_1$ implies $\bar{x}_2$" and "x_2 implies x_1". Construct a digraph with $2n$ nodes, where the first n nodes represent $x_1, \ldots, x_n$ and the others represent $\bar{x}_1, \ldots, \bar{x}_n$. Represent every implication by a directed arc. For example, the first of the above implications is represented by an arc from node $\bar{x}_1$ to node $\bar{x}_2$. Compute the strongly connected components of the digraph. The instance is satisfiable if and only there does not exist an index i such that both node x_i and node $\bar{x}_i$ lie in the same component.

2.3 Gadgets

Gadgets (also sometimes called *units* or *components*) are used heavily in computational complexity theory, in order prove hardness results. For example, suppose we wish to prove that `3-SAT` is $\mathcal{NP}$-complete. Since we know that SAT is $\mathcal{NP}$-complete, it suffices to show that any clause with more than 3 terms is equivalent to a small (polynomially bounded) number of clauses with only 3 terms (possibly with the help of additional variables). This is indeed the case. For example, suppose we have a clause of the form $\bigvee_{i=1}^{p} x_i$, with $p \geq 4$. Let $q = \lceil p/2 \rceil$ and replace the clause with two shorter clauses of the form $y \vee \bigvee_{i=1}^{q} x_i$ and $\bar{y} \vee \bigvee_{i=q+1}^{p} x_i$, where y is an additional Boolean variable. Using this "gadget" repeatedly, if necessary, we eventually obtain $O(p)$ clauses with only three terms each, using only $O(p)$ additional variables. Instead of giving more details, we refer the reader to [19, 38].

2.4 The ATSP, SSP and CPP

Finally, we recall some facts about the ATSP, SSP and CPP.

Given an integer $n \geq 2$ and a cost $c_a \in \mathbb{Q}$ for each $a \in A_n$, the ATSP is the problem of finding a minimum-cost Hamiltonian dicycle in D_n. The standard ILP formulation of the ATSP is as follows [15]. For each arc $a \in A$, let x_a be a binary variable, taking the value 1 if and only if the arc a is traversed. Then:

$$\begin{array}{llll} \max & \sum_{a \in A} c_a x_a & & \\ \text{s.t.} & \sum_{j \neq i} x_{ij} = 1 & (i \in V_n) & (1) \\ & \sum_{j \neq i} x_{ji} = 1 & (i \in V_n) & (2) \\ & \sum_{i \in S, j \in S \setminus \{i\}} x_{ij} \leq |S| - 1 & (S \subset V_n : 2 \leq |S| \leq n/2) & (3) \\ & x_a \in \{0,1\} & (a \in A_n). & \end{array}$$

The constraints (1) and (2) are called *out-* and *in-degree* equations. The constraints (3) are called *subtour elimination constraints* (SECs).

Many facet-defining inequalities are known for the ATSP; see, e.g., [3, 33] for surveys. It is known that the separation problem for the SECs can be solved in $O(n^3)$ time [34]. Polynomial-time separation algorithms for other families of inequalities can be found in [6, 7, 17, 27, 28, 36]. One family that will be of interest to us is the *weak odd closed alternating trail* (weak odd CAT) inequalities [2, 6]. These are in fact nothing but the inequalities that can be derived as $\{0, \frac{1}{2}\}$-cuts from inequalities of the form $x_{ij} + x_{ji} \leq 1$, $x_{ij} + x_{ik} \leq 1$ and $x_{ij} + x_{kj} \leq 1$. From this it follows that the associated separation problem can be solved exactly in $O(n^5)$ time [6].

Given an undirected graph $G = (V, E)$, the SSP is the problem of finding a stable set of maximum cardinality. The standard ILP formulation of the SSP is as follows [22]. For each $i \in V$, let x_i be a binary variable, taking the value 1 if and only if node i is selected. Then:

$$\begin{array}{llll} \max & \sum_{i \in V} x_i & & \\ \text{s.t.} & x_i + x_j \leq 1 & (\{i,j\} \in E) & (4) \\ & x_i \in \{0,1\} & (i \in V). & \end{array}$$

The constraints (4) are called *edge inequalities*.

Many strong valid inequalities are known for the SSP (see the surveys in [21, 22]). For example, Padberg [35] introduced the following inequalities (among others):

- *clique* inequalities $\sum_{i \in C} x_i \leq 1$ for each maximal clique $C \subseteq V$;
- *odd hole* inequalities $\sum_{i \in H} x_i \leq \lfloor |H|/2 \rfloor$ for each set of nodes H inducing an odd hole (chordless cycle of odd length) in G;
- *odd antihole* inequalities $\sum_{i \in A} x_i \leq 2$ for each set of nodes A inducing the complement of an odd hole in G.

Note that the clique inequalities dominate the edge inequalities.

The separation problem for odd hole inequalities can be reduced to a minimum-weight odd cycle problem [4, 20], and thereby solved efficiently. In fact, the odd hole inequalities are $\{0, \frac{1}{2}\}$-cuts with respect to the edge inequalities, and the approaches described in Subsection 2.1 include the algorithm in [4, 20] as a special case. Polynomial-time separation algorithms for other SSP inequalities are given in [5, 8, 9, 21, 22, 29].

Finally, given an integer $n \geq 2$ and a rational weight w_e for each edge $e \in E_n$, the CPP calls for a partition of V_n into subsets (cliques), such that the sum of the weights of the edges that have both end-nodes in the same clique is maximised [30]. The standard ILP formulation of the CPP is the following [24, 30]. For all $\{i, j\} \in E$, the binary variable x_{ij} takes the value 1 if and only if nodes i and j are in the same clique. Then:

$$\begin{array}{lll} \max & \sum_{e \in E_n} w_e x_e & \\ \text{s.t.} & x_{ik} + x_{jk} - x_{ij} \leq 1 & (\{i,j\} \in E_n,\ k \in V_n \setminus \{i,j\}) \\ & x_e \in \{0,1\} & (e \in E_n). \end{array} \qquad (5)$$

The inequalities (5) are called *transitivity* inequalities.

Grötschel and Wakabayashi [24] showed that the transitivity inequalities (5) define facets. They also introduced several other families of facet-defining inequalities. Of those, we will be especially interested in the *2-chorded odd cycle* (2-COC) inequalities. These take the form:

$$\sum_{e \in E(C)} x_e - \sum_{e \in H} x_e \leq \lfloor |C|/2 \rfloor,$$

where C is a cycle in K_n with $|C| \geq 5$ and odd, and H is the set of *2-chords* of C, i.e., the set of edges in E_n which connect pairs of nodes in $V(C)$ that have a distance 2 in C.

Chopra and Rao [10] found two additional families of facet-defining inequalities for the CPP. Of those, we will be interested in the *odd wheel* (OW) inequalities. These take the form:

$$\sum_{v \in V(C)} x_{vh} - \sum_{e \in E(C)} x_e \leq \lfloor |C|/2 \rfloor,$$

where C is a cycle in K_n with $|C| \geq 3$ and odd, and h is any arbitrary node in $V_n \setminus V(C)$.

Deza *et al.* [16] showed that the separation problem for the OW inequalities can be solved in $O(n^4)$ time. The complexity of separation for the 2-COC inequalities is unknown, but it was shown in [6, 31] that there exists an $O(n^5)$-time separation algorithm for a family of valid inequalities that includes all OW and 2-COC inequalities. These inequalities, called *odd closed walk* (OCW) inequalities, are derived as $\{0, \frac{1}{2}\}$-cuts from weakened transitivity inequalities of the form $x_{ij} + x_{jk} - 2x_{ik} \leq 1$.

3 Gadgets, Cutting Planes and Separation Algorithms

In this section, we present our new approach. In Subsections 3.1 and 3.2, we define gadgets and explain how they can be used to derive cuts. In Subsection 3.3, we present a method for constructing gadgets for 0-1 linear programs (0-1 LPs). In Subsection 3.4, we show how to use gadgets to derive efficient separation algorithms for 0-1 LPs. Finally, in Subsection 3.5, we make some additional remarks.

Throughout this section, we assume that our COP has been formulated as an ILP with n variables. We also let $S \subset \mathbb{Z}^n_+$ denote the set of feasible solutions and let $P = \text{conv}(S)$ be the associated polytope.

3.1 Gadgets

We now define our gadgets formally. Let us assume that we can identify some interesting "properties" that a feasible solution can have. Such properties could be very simple (e.g., "x_j takes the value 0", "x_j takes an even value"), or more complex (e.g., "the sum of x_j and x_k is a prime number", "the solution corresponds to a forest"). Given a specific feasible solution to the ILP, we write "h_p" when the solution has property p, and "$\bar{h}_p$" when it does not. We can then write clauses, exactly as in the satisfiability problem. For example, "$h_p \vee \bar{h}_q$" means "the solution has property p but does not have property q".

We will need the following two definitions.

Definition 1 *Let $\alpha^T x \leq \beta$ be any linear inequality that is valid for P. A point $x \in S$ satisfying $\alpha^T x = \beta$ will be called a "root" of the inequality.*

Definition 2 *A valid inequality $\alpha^T x \leq \beta$ will be called a "gadget" (with respect to a given ILP and a given collection of properties) if it satisfies the following conditions:*

- *it has integral coefficients, i.e. $\alpha \in \mathbb{Z}^n$ and $\beta \in \mathbb{Z}$;*
- *it has at least one root;*
- *at least one clause is known that is satisfied by all roots.*

For clarity, we give an example.

Example 1 *Let $S = \left\{x \in \mathbb{Z}^3_+ : x_1 + x_2 + 3x_3 = 7\right\}$. Suppose our properties are "x_1 is odd", "x_2 is odd" and "x_3 is odd". The inequality $x_1 \geq 0$ is (trivially) valid for P. The roots of the inequality are $(0, 7, 0)$, $(0, 4, 1)$ and $(0, 1, 2)$. In all roots, exactly one of the last two variables is odd. So, all roots satisfy the clauses $h_2 \vee h_3$ and $\bar{h}_2 \vee \bar{h}_3$. Thus, the given inequality is a gadget. One can check that the valid inequality $x_3 \leq 2$ is also a gadget, since all roots satisfy the clauses $h_1 \vee h_2$ and $\bar{h}_1 \vee \bar{h}_2$.* □

3.2 Using gadgets to derive cuts

Next, we present a simple way to use gadgets to generate cutting planes.

Lemma 1 *Suppose we are given an ILP, a collection of properties, and a collection R of gadgets, of the form $\alpha^r \cdot x \leq \beta^r$ for $r \in R$. Let R' be a subset of R. We construct an instance of* `SAT`*, simply by including all clauses associated with the gadgets in R'. If the* `SAT` *instance is unsatisfiable, then the inequality*

$$\sum_{r \in R'} \alpha^r \cdot x \leq \sum_{r \in R'} \beta^r - 1$$

is valid for P.

Proof. Given that the `SAT` instance is unsatisfiable, it is impossible for a feasible solution to satisfy all of the gadgets in R' at equality simultaneously. Thus, at least one of them must have a positive slack. Given that gadgets have integral coefficients, the sum of the slacks must be at least one. □

Unfortunately, this lemma is too general to be of practical use, given that `SAT` is $\mathcal{NP}$-complete in the strong sense. Moreover, in itself, it does not yield an efficient separation algorithm.

3.3 Gadgets for 0-1 LPs

We now focus on gadgets for 0-1 LPs. For simplicity and brevity, we assume that we have only n properties, of the form "$x_i = 1$" for $i = 1, \ldots, n$. We also define two special kinds of gadget:

Definition 3 *A gadget will be called an "exclusive or" (*`XOR`*) gadget if there exists a pair $\{i, j\} \subset \{1, \ldots, n\}$ such that, in all roots, $x_i + x_j = 1$.*

Definition 4 *A gadget will be called an "equality" (*`EQ`*) gadget if there exists a pair $\{i, j\} \subset \{1, \ldots, n\}$ such that, in all roots, $x_i = x_j$.*

The following example shows that it is possible for a gadget to be an `XOR` gadget and an `EQ` gadget simultaneously.

Example 2 *Suppose that $S \in \{0, 1\}^5$, and that the inequality $x_1+x_2+2x_3+3x_4+5x_5 \leq 6$ is valid for P. All roots of this inequality satisfy $x_1 + x_2 = 1$, $x_3 = x_4$, $x_3+x_5 = 1$ and $x_4+x_5 = 1$. Thus, the inequality is an* `XOR` *gadget with respect to $\{1, 2\}$, $\{3, 5\}$ and $\{4, 5\}$, but an* `EQ` *gadget with respect to $\{3, 4\}$.* □

The following lemma presents some simple `XOR` and `EQ` gadgets.

Lemma 2 *If the inequality $x_i + x_j \leq 1$ or $x_i + x_j \geq 1$ is valid for P, then it is an* `XOR` *gadget. If the inequality $x_i \leq x_j$ is valid for P, then it is an* `EQ` *gadget.*

Proof. Trivial. □

The following theorem presents a general procedure for generating more interesting `XOR` and `EQ` gadgets.

Theorem 1 *Let $\alpha^T x \leq \beta$ be an arbitrary valid inequality for P that (a) has integral coefficients and (b) has at least four roots. We say that a pair $\{i, j\} \subset \{1, \ldots, n\}$ is "compatible" with the given inequality if, for any pair $s, t \in \{0, 1\}$, there exists at least one root satisfying $x_i = s$ and $x_j = t$. Assume that $\{i, j\}$ is compatible. Define:*

$$\beta^{st} = \max\left\{\alpha^T x : x \in S, x_i = s, x_j = t\right\},$$

and let $\Delta = \beta^{10} + \beta^{01} - \beta^{00} - \beta^{11}$. If $\Delta = 1$, then the inequality

$$2\alpha^T x + \left(2\beta^{00} - 2\beta^{10} + 1\right)x_i + \left(2\beta^{00} - 2\beta^{01} + 1\right)x_j \leq 2\beta^{00} + 1 \qquad (6)$$

is an `XOR` *gadget. If $\Delta \geq 2$, then the inequality*

$$\alpha^T x + \left(\beta^{00} - \beta^{10} + 1\right) x_i + \left(\beta^{00} - \beta^{01} + 1\right) x_j \leq \beta^{00} + 1$$

is an XOR *gadget. If* $\Delta \geq 3$*, then the inequality*

$$\alpha^T x + \big(\beta^{01} - \beta^{11} - 1\big)x_i + \big(\beta^{10} - \beta^{11} - 1\big)x_j \leq \beta^{00} + \Delta - 1$$

is also an XOR *gadget. If* $\Delta = -1$*, then the inequality*

$$2\alpha^T x + \big(2\beta^{00} - 2\beta^{10} - 1\big)x_i + \big(2\beta^{00} - 2\beta^{01} - 1\big)x_j \leq 2\beta^{00}$$

is an EQ *gadget. If* $\Delta = -2$*, then the inequality*

$$\alpha^T x + \big(\beta^{00} - \beta^{10} - 1\big)x_i + \big(\beta^{00} - \beta^{01} - 1\big)x_j \leq \beta^{00}$$

is an EQ *gadget. Finally, if* $\Delta \leq -3$*, then the inequalities*

$$\begin{aligned} \alpha^T x + \big(\beta^{00} - \beta^{10} - 1\big)x_i + \big(\beta^{10} - \beta^{11} + 1\big)x_j &\leq \beta^{00} \\ \alpha^T x + \big(\beta^{01} - \beta^{11} + 1\big)x_i + \big(\beta^{00} - \beta^{01} - 1\big)x_j &\leq \beta^{00} \end{aligned}$$

are EQ *gadgets.*

Proof. For brevity, we present the proof only for the inequality (6). (The proof is similar for the other inequalities, but more tedious.) To show that (6) is valid for P, we consider four cases:

Case 1: $x_i = x_j = 0$. In this case, (6) reduces to $2\alpha^T x \leq 2\beta^{00} + 1$. Validity here follows from the definition of β^{00}.
Case 2: $x_i = x_j = 1$. Now, (6) reduces to $2\alpha^T x \leq 2(\beta^{10} + \beta^{01} - \beta^{00}) = 2\beta^{11} + 1$. Validity here follows from the definition of β^{11}.
Case 3: $x_i = 0$ and $x_j = 1$. Now, (6) reduces to $2\alpha^T x \leq 2\beta^{01}$. Validity here follows from the definition of β^{01}.
Case 4: $x_i = 1$ and $x_j = 0$. Now, (6) reduces to $2\alpha^T x \leq 2\beta^{10}$. Validity here follows from the definition of β^{10}.

Now observe that, in the first two cases, no roots can exist, since all left-hand side coefficients are even and the right-hand side is odd. Thus, (6) is an XOR gadget. □

3.4 Cuts and separation for 0-1 problems

Now we use XOR and EQ gadgets to define a broad family of cutting planes for 0-1 LPs.

Proposition 1 (Odd Gadget Cycle Inequalities) *Consider an arbitrary collection* R *of* XOR *and* EQ *gadgets. Let the gadgets be written in the form* $\alpha^r \cdot x \leq \beta^r$ *for* $r \in R$*. Construct a multigraph, say* $G^+ = (V^+, E^+)$*, as follows.*

- *Set* V^+ *to* $\{1, \ldots, n\}$*.*

- *For all $r \in R$ and $\{i,j\} \subset V^+$, if gadget r is an* `XOR` *gadget with respect to $\{i,j\}$, insert an edge into E^+ between i and j, and label the edge "odd".*

- *For all $r \in R$ and $\{i,j\} \subset V^+$, if gadget r is an* `EQ` *gadget with respect to $\{i,j\}$, insert an edge into E^+ between i and j, and label the edge "even".*

Let C be any odd cycle in G^+. Let $r(1), \ldots, r(t)$ be the gadgets that correspond to the edges in $E(C)$. The "odd gadget cycle" (OGC) inequality

$$\sum_{s=1}^{t} \alpha^{r(s)} \cdot x \leq \sum_{s=1}^{t} \beta^{r(s)} - 1 \tag{7}$$

is valid for P.

Proof. Suppose that a solution $\tilde{x} \in S$ is a root of all t gadgets simultaneously. If $\{i,j\} \in E(C)$ is an odd edge, then $\tilde{x}_i \neq \tilde{x}_j$. If it is an even edge, then $\tilde{x}_i = \tilde{x}_j$. Since $E(C)$ contains an odd number of odd edges, we have $\tilde{x}_i \neq \tilde{x}_i$ for all $i \in V(C)$. Since this is a contradiction, at least one of the t gadgets must have a positive slack. □

We will see in the following three sections that, for some COPs and some simple collections of gadgets, the number of non-dominated OGC inequalities can be exponential in n. Fortunately, the separation problem for the OGC inequalities can be solved exactly in polynomial time, as we now show.

Proposition 2 (Separation of OGC Inequalities) *If R is a collection of* `XOR` *and* `EQ` *gadgets whose size is bounded by a polynomial in n, then the separation problem for the corresponding OGC inequalities can be solved in polynomial time.*

Proof. Let $x^* \in [0,1]^n$ be the point to be separated. If any of the gadgets themselves are violated, then we immediately have a cutting plane. So assume that x^* satisfies all of the given gadgets. Construct the multigraph G^+, and set the weight of each edge to the slack of the associated gadget. Note that all edges have non-negative weight. By construction, there is a one-to-one correspondence between violated OGC inequalities and odd cycles in G^+ whose total weight is less than 1. We can find a minimum-weight odd cycle in polynomial time using the approach in [4, 20]. □

3.5 Additional remarks

To end this section, we make some additional remarks about OGC inequalities.

1. Any odd cycle in G^+ with weight less than 1 represents a violated OGC inequality. As a result, the separation algorithm can potentially find up to n violated inequalities in a single call.

2. It may be that two or more edges in an odd cycle represent the same gadget. In this case, one can strengthen the OGC inequality (7) by including just one copy of each distinct gadget in the summation. We suspect that the separation problem for these strengthened OGC inequalities is $\mathcal{NP}$-hard. In practice, one can just take each (near-)violated OGC inequality found by the odd cycle approach, and check whether it can be strengthened.

3. If G^+ has parallel edges between any pair of nodes, we can omit all apart from two; namely, the odd and even edges with smallest weight. So we can assume that G^+ contains $O(n^2)$ edges. From this it follows that the minimum-weight odd cycle can be found in $O(n^3)$ time. The total separation time could be larger or smaller, however, depending on the number of gadgets and the time taken to compute their slacks.

4. A constraint with two odd left-hand side coefficients is either an `XOR` gadget or an `EQ` gadget, depending on whether the right-hand side is odd or even. From this it can be shown that any $\{0, \frac{1}{2}\}$-cut that can be obtained with the Caprara–Fischetti approach can also be obtained by dividing an OGC inequality by two. Thus, the Caprara–Fischetti approach is a special case of ours. Moreover, our approach is more general, since we do not impose restrictions on the parities of the coefficients in the gadgets. (Indeed, the gadget in Example 2 has more than two odd left-hand side coefficients.)

4 Application to Specific COPs

In this section, we apply the gadget approach to the ATSP, SSP and CPP.

4.1 Application to the ATSP

We begin with the ATSP. The following lemma presents three very simple `XOR` gadgets.

Lemma 3 *The following inequalities are* `XOR` *gadgets for the ATSP:*

$$x_{ij} + x_{ji} \leq 1 \qquad (i, j \in V_n)$$
$$x_{ij} + x_{ik} \leq 1 \qquad (i, j, k \in V_n)$$
$$x_{ij} + x_{kj} \leq 1 \qquad (i, j, k \in V_n).$$

Proof. Trivial. □

With the aid of Theorem 1, we have found several more complex gadgets. Some examples are given in the following two propositions.

Proposition 3 *Let (i, j, k) be an ordered subset of V_n. The following inequalities are* `XOR` *gadgets for the ATSP, with respect to the variables x_{ij} and x_{jk}:*

$$
\begin{aligned}
x_{ij} + x_{jk} + 2x_{ik} + 2x_{ki} &\le 3 \\
x_{ij} + x_{jk} + 2x_{ji} + 2x_{ki} &\le 3 \\
x_{ij} + x_{jk} + 2x_{ki} + 2x_{kj} &\le 3 \\
x_{ij} + x_{jk} + x_{ik} + x_{ji} + x_{kj} + 2x_{ki} &\le 3 \\
x_{ij} + x_{jk} + 2x_{ik} + x_{ki} + x_{i\ell} + x_{\ell k} &\le 3 \qquad (\ell \in V_n \setminus \{i, j, k\}) \\
x_{ij} + x_{jk} + 2 \sum_{p,q \in \{i,k,\ell\}} x_{pq} &\le 5 \qquad (\ell \in V_n \setminus \{i, j, k\}) \\
x_{ij} + x_{jk} + 2 \sum_{p \in \{i,j,k\}} (x_{\ell p} + x_{p\ell}) &\le 5 \qquad (\ell \in V_n \setminus \{i, j, k\}).
\end{aligned}
$$

Proof. One can check (e.g., by brute-force enumeration) that (a) these inequalities are valid and (b) in any root, $x_{ij} + x_{jk} = 1$. □

Proposition 4 *Let (i, j, k) be an ordered subset of V_n. The following inequalities are* `EQ` *gadgets for the ATSP, with respect to the variables x_{ij} and x_{jk}:*

$$
\begin{aligned}
x_{ij} + x_{jk} + x_{ik} + x_{ji} + x_{kj} &\le 2 \\
2x_{ij} + 2x_{jk} + 2x_{ik} + x_{ji} + 2x_{kj} + x_{i\ell} + x_{\ell k} &\le 4 \qquad (\ell \in V_n \setminus \{i, j, k\}) \\
2x_{ij} + 2x_{jk} + 2x_{ik} + 2x_{ji} + x_{kj} + x_{i\ell} + x_{\ell k} &\le 4 \qquad (\ell \in V_n \setminus \{i, j, k\}) \\
\sum_{p,q \in \{i,j,k,\ell\}} x_{pq} + x_{kj} + x_{k\ell} - x_{\ell i} &\le 4 \qquad (\ell \in V_n \setminus \{i, j, k\}) \\
\sum_{p,q \in \{i,j,k,\ell\}} x_{pq} + x_{ji} + x_{\ell i} - x_{k\ell} &\le 4 \qquad (\ell \in V_n \setminus \{i, j, k\}).
\end{aligned}
$$

Proof. One can check (e.g., by brute-force enumeration) that (a) these inequalities are valid and (b) in any root, $x_{ij} = x_{jk}$. □

We remark that two of our `XOR` gadgets and all five of our `EQ` gadgets have more than two odd coefficients on the left-hand side.

One can check that, even if one uses only the `XOR` gadgets described in Lemma 3, the OGC inequalities include all weak odd CAT inequalities. Adding the other gadgets to the collection, one can derive many other interesting OGC inequalities for the ATSP. Here is an example.

Example 3 *From Proposition 3, the following inequality is an* `XOR` *gadget with respect to* x_{12} *and* x_{23}*:*

$$x_{12} + x_{23} + x_{13} + x_{21} + x_{32} + 2x_{31} \leq 3.$$

Moreover, from Lemma 3, the following four inequalities are also `XOR` *gadgets:* $x_{23} + x_{43} \leq 1$, $x_{43} + x_{45} \leq 1$, $x_{45} + x_{15} \leq 1$ *and* $x_{15} + x_{12} \leq 1$*. One can check that these five gadgets correspond to an odd cycle in* G^+*, passing through the nodes* $(1,2)$, $(2,3)$, $(4,3)$, $(4,5)$ *and* $(1,5)$*. This yields the OGC inequality*

$$2\big(x_{12} + x_{15} + x_{23} + x_{43} + x_{45}\big) + x_{13} + x_{21} + x_{32} + x_{31} \leq 6.$$

This inequality cuts off fractional points with $x_{31} = 1$ *and* $x_{12} = x_{23} = x_{43} = x_{45} = x_{15} = 1/2$*. One can check that such points cannot be cut off with SECs or weak odd CAT inequalities.* □

Now consider the separation problem associated with the OGC inequalities that can be derived using our gadgets. One can check that the multigraph G^+ has $O\big(n^2\big)$ nodes and $O\big(n^3\big)$ edges. From this one can show that the separation algorithm takes $O\big(n^5\big)$ time. A more careful analysis enables one to reduce the time to $O\big(fn^2 + f^2 n\big)$, where f is the number of variables that take a fractional value in the given fractional solution. We omit details for brevity. One can also show that the algorithm can return up to f violated inequalities in a single call.

4.2 Application to the SSP

We now apply the gadget approach to the SSP. The following lemma presents some very simple `XOR` gadgets.

Lemma 4 *Every edge inequality (4) is an* `XOR` *gadget for the SSP.*

Proof. Trivial. □

With the aid of Theorem 1, we have found several more complex gadgets. Some examples are given in the following two propositions.

Proposition 5 *Let* i *and* j *be a pair of non-adjacent nodes in* G*. If* $C \subset V \setminus \{i,j\}$ *is a clique such that (a) each node in* C *is adjacent to exactly one of* i *and* j*, (b)* $|C \cap n(i)| \geq 2$ *and (c)* $|C \cap n(j)| \geq 2$*, then the inequality*

$$x_i + x_j + 2\sum_{k \in C} x_k \leq 3$$

is an `XOR` *gadget with respect to* x_i *and* x_j*.*

Proposition 6 *Let i and j be a pair of non-adjacent nodes in G. If C be a maximal subset of $n(i) \cap n(j)$ such that (a) C is a clique and (b) $|C| \geq 2$, then the inequality*

$$x_i + x_j + 2 \sum_{k \in C} x_k \leq 2 \tag{8}$$

is an `EQ` *gadget with respect to x_i and x_j. Moreover, if A is a subset of $n(i) \cap n(j)$ that induces an odd antihole in G, then the inequality*

$$x_i + x_j + \sum_{k \in A} x_k \leq 2 \tag{9}$$

is an `EQ` *gadget with respect to x_i and x_j.*

We remark that the gadget (9) does not have two odd left-hand side coefficients.

One can check that, if we use only the edge inequalities (4) as `XOR` gadgets, the OGC inequalities are precisely the odd hole inequalities. Adding the other gadgets to the collection, one can derive many other interesting OGC inequalities for the SSP. Here are two examples.

Example 4 *Let G be the graph in Fig. 1. Suppose we use the* `XOR` *gadget $x_1 + x_7 \leq 1$ and the following two* `EQ` *gadgets of type (8):*

$$x_1 + x_4 + 2\big(x_2 + x_3 + x_8\big) \leq 2$$
$$x_4 + x_7 + 2\big(x_5 + x_6 + x_8\big) \leq 2.$$

One can check that these three gadgets correspond to an odd cycle in G^+, passing through nodes 1, 4 and 7. This yields the OGC inequality

$$2 \sum_{i=1}^{7} x_i + 4x_8 \leq 4.$$

Dividing this by two, we obtain $\sum_{i=1}^{7} x_i + 2x_8 \leq 2$. This can be shown to be facet-defining by applying Padberg's "lifting" procedure [35] to the odd hole inequality $x_1 + x_3 + x_4 + x_5 + x_7 \leq 2$. □

Example 5 *Let G be the (unique) graph with 13 nodes and 31 edges such that:*

- $\{1, \ldots, 5\}$ *and* $\{6, \ldots, 10\}$ *induce 5-antiholes (which are also 5-holes);*
- *node 11 is adjacent to nodes $1, \ldots, 10$;*
- *node 12 is adjacent to nodes $1, \ldots, 5$;*

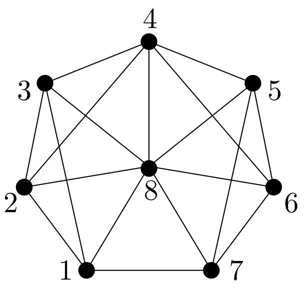

Figure 1: Graph for Example 4.

- *node* 13 *is adjacent to nodes* $6, \ldots, 10$*;*
- *nodes* 12 *and* 13 *are adjacent.*

(See Fig. 2.) Suppose we use the `XOR` *gadget* $x_{12} + x_{13} \leq 1$ *and the following two* `EQ` *gadgets of type (9):*

$$x_{11} + x_{12} + \sum_{i=1}^{5} x_i \leq 2$$

$$x_{11} + x_{13} + \sum_{i=6}^{10} x_i \leq 2.$$

One can check that these three gadgets correspond to an odd cycle in G^+*, passing through nodes* 11*,* 12 *and* 13*. This yields the OGC inequality*

$$2\big(x_{11} + x_{12} + x_{13}\big) \; + \; \sum_{i=1}^{10} x_i \; \leq \; 4.$$

One can check (either by hand, or with the help of a software package such as `PORTA` *[11]) that this inequality defines a facet. One can also check that* G *does not contain any holes of cardinality 9, which means that the OGC inequality is not a lifted odd hole inequality.* □

Now consider the separation problem associated with the OGC inequalities that can be derived using our gadgets. One can check that:

- If we include all gadgets involving four nodes or fewer, the separation algorithm takes $O\big(|V|^3 + |E|^2\big)$ time.
- If we also include all gadgets involving five nodes, the time increases to $O\big(|V|\,|E|^2\big)$.
- If we also include all gadgets involving six nodes, the time increases to $O\big(|E|^3\big)$.

As before, one can obtain improved running times by considering the variables that take fractional values. We omit details for brevity.

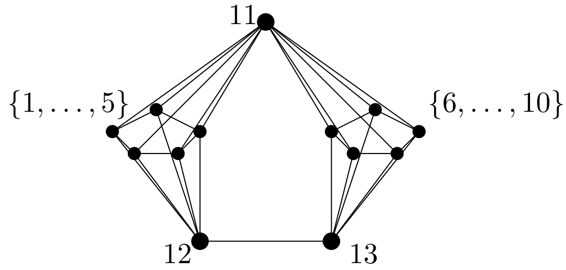

Figure 2: Graph for Example 5.

4.3 Application to the CPP

Finally, we apply the gadget approach to the CPP. The following lemmas describe two simple, yet remarkably powerful, gadgets for the CPP.

Lemma 5 *For any $\{i,j\} \in E_n$ and any $k \in V_n \setminus \{i,j\}$, the inequality*

$$x_{ik} + x_{jk} - 2x_{ij} \leq 1 \tag{10}$$

is an `XOR` *gadget for the CPP.*

Proof. Validity is trivial. One can check that in all roots, exactly one of the variables x_{ik} and x_{jk} is equal to one. □

Lemma 6 *For any $\{i,j\} \in E_n$, the inequality*

$$x_{ij} \leq 1 \tag{11}$$

is an `EQ` *gadget for the CPP. This gadget is associated with $n-2$ distinct pairs of variables.*

Proof. Validity is again trivial. If x is a root, then nodes i and j must lie in the same clique. This implies that, for any node $k \in V \setminus \{i,j\}$, the variables x_{ik} and x_{jk} must take the same value. □

The following proposition shows that the `XOR` gadgets (10) alone are rather powerful.

Proposition 7 *Every OGC inequality derived from the* `XOR` *gadgets (10) is equivalent to an OCW inequality and vice-versa.*

Proof. Let G^+ be the graph obtained when R consists of the `XOR` gadgets. Note that there is one node in G^+ for each edge $e \in E_n$. Consider an odd cycle in G^+, and let c denote the number of edges in the cycle. Let $e(1), \ldots, e(c)$ be the edges in E_n that correspond to the c nodes in the cycle.

Also, for notational simplicity, let $e(c+1) = e(1)$. The ith edge in the cycle then corresponds to the `XOR` gadget $x_{e(i)} + x_{e(i+1)} - 2x_{f(i)} \leq 1$, where $f(i)$ is the edge having one end-node in common with $e(i)$ and $e(i+1)$. The OGC inequality can therefore be written as

$$\sum_{i=1}^{c} \left(x_{e(i)} + x_{e(i+1)} \; - \; 2x_{f(i)} \right) \; \leq \; c - 1.$$

Now observe that each edge $e(i)$ is counted twice on the left-hand side. Thus, if we divide the OGC inequality by two, we obtain:

$$\sum_{i=1}^{c} x_{e(i)} \; - \; \sum_{i=1}^{c} x_{f(i)} \; \leq \; \lfloor c/2 \rfloor.$$

This is an OCW inequality. Similarly, given any OCW inequality, one can construct an odd cycle in G^+ that yields a OGC inequality that is twice the given OCW inequality. □

It turns out that, if we use the `XOR` gadgets (10) and the `EQ` gadgets (11) in combination, we can obtain facet-defining OGC inequalities that are *not* OCW inequalities. Here are two examples.

Example 6 *Suppose we use the* `XOR` *gadgets*

$$x_{12} + x_{13} - 2x_{23} \leq 1, \;\; x_{13} + x_{14} - 2x_{34} \leq 1, \;\; x_{14} + x_{15} - 2x_{45} \leq 1;$$

and the `EQ` *gadget* $x_{25} \leq 1$*. One can check that these four gadgets correspond to an odd cycle in* G^+*, passing through the nodes* $\{1,2\}$*,* $\{1,3\}$*,* $\{1,4\}$ *and* $\{1,5\}$*. This yields the OGC inequality.*

$$x_{12} + x_{15} + x_{25} + 2\big(x_{13} + x_{14} - x_{23} - x_{34} - x_{45}\big) \leq 3. \qquad (12)$$

This can be shown to define a facet when $n = 5$*.* □

Example 7 *Suppose we use the* `XOR` *gadgets*

$$x_{17} + x_{27} - 2x_{12} \leq 1, \;\; x_{27} + x_{37} - 2x_{23} \leq 1, \;\; x_{37} + x_{47} - 2x_{34} \leq 1,$$

$$x_{47} + x_{57} - 2x_{45} \leq 1, \;\; x_{57} + x_{67} - 2x_{56} \leq 1,$$

and the `EQ` *gadget* $x_{16} \leq 1$*. One can check that these six gadgets correspond to an odd cycle in* G^+*, passing through the nodes* $\{1,7\}$*,* $\{2,7\}$*,* $\{3,7\}$*,* $\{4,7\}$*,* $\{5,7\}$ *and* $\{6,7\}$*. This yields the OGC inequality*

$$x_{16}+x_{17}+x_{67}+2\big(x_{27}+x_{37}+x_{47}+x_{57}-x_{12}-x_{23}-x_{34}-x_{45}-x_{56}\big) \leq 5. \qquad (13)$$

This can be shown to define a facet when $n = 7$*.* □

Readers who are familiar with the concept of *Chvátal rank* (see [12]) may find the following result of interest.

Proposition 8 *The OGC inequalities (12) and (13) have Chvátal rank larger than* 1*.*

Proof. Since the CPP polytope is full-dimensional and the inequalities in question define facets for $n = 5$ and $n = 7$, respectively, we can use the method in [25]. When $n = 5$, if we maximise the left-hand side of the inequality (12) subject to the transitivity and non-negativity inequalities, we obtain a fractional LP solution with $x_{25} = 1$, $x_{12} = x_{13} = x_{14} = x_{15} = 1/2$, and all other variables equal to zero. The profit of this solution is 4. Since the right-hand side of (12) is $3 < \lfloor 4 \rfloor$, the inequality must have rank larger than 1.

Similarly, when $n = 7$, if we maximise the left-hand side of the inequality (13) subject to the transitivity and non-negativity inequalities, we obtain a fractional solution with $x_{16} = 1$ and $x_{i7} = 1/2$ for $i = 1, \ldots, 6$. The profit of this solution is 6, whereas the right-hand side of (13) is $5 < \lfloor 6 \rfloor$. □

In particular, the OGC inequalities (12) and (13) are not implied by $\{0, \frac{1}{2}\}$-cuts.

As for the separation problem for our OGC inequalities, the situation is similar to that of the ATSP. That is, the separation algorithm takes $O(n^5)$ time, but a more careful analysis enables one to reduce the time to $O(fn^2 + f^2n)$, where f is the number of fractional variables. We omit details for brevity.

5 Conclusions

Up to now, gadgets have been used solely to prove hardness results. In this paper, we have shown that they can also be useful for deriving new families of cutting planes, with accompanying efficient separation algorithms. The results that we obtained with the asymmetric traveling salesman, stable set and clique partitioning problems suggest that this new approach has considerable potential.

We can think of three interesting topics for future research. The first is whether the gadget approach can be usefully applied to other $\mathcal{NP}$-hard combinatorial optimisation problems. Problems worth consideration include, e.g., the symmetric TSP, the max-cut problem, the *linear ordering problem* [23] and the *transitive acyclic subdigraph problem* [31]. The second is the possibility of defining gadgets that impose more complex structural properties on the roots, rather than merely imposing relationships between the parities of individual pairs of variables. The third, suggested by an anonymous referee, is the possibility of generating gadgets "on-the-fly", based on the structure of the current fractional point to be separated.

Acknowledgement: The authors would like to thank the anonymous referee and the associate editor for very helpful suggestions.

References

[1] B. Aspvall, M.F. Plass & R.E. Tarjan (1979) A linear-time algorithm for testing the truth of certain quantified boolean formulas. *Inform. Proc. Lett.*, 8, 121–123.

[2] E. Balas (1989) The asymmetric assignment problem and some new facets of the traveling salesman polytope on a directed graph. *SIAM J. Discr. Math.*, 2, 425–451.

[3] E. Balas & M. Fischetti (2002) Polyhedral theory for the asymmetric traveling salesman problem. In G. Gutin & A.P. Punnen (eds.) *The Traveling Salesman Problem and Its Variations*, pp. 117–168. Dortrecht: Kluwer.

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

[5] R. Borndörfer & R. Weismantel (2000) Set packing relaxations of integer programs. *Math. Program.*, 88, 425–450.

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

[7] R.D. Carr (1997) Separating clique tree and bipartition inequalities having a fixed number of handles and teeth in polynomial time. *Math. Oper. Res.*, 22, 257–265.

[8] E. Cheng & W.H. Cunningham (1997) Wheel inequalities for stable set polytopes. *Math. Program.*, 77, 389–421.

[9] E. Cheng & S. de Vries (2002). Antiweb-wheel inequalities and their separation problems over the stable set polytopes. *Math. Program.*, 92, 153–175.

[10] S. Chopra & M.R. Rao (1993) The partition problem. *Math. Program.*, 59, 87–115.

[11] T. Christof & A. Loebl, PORTA (polyhedron representation transformation algorithm). Software package, available for download at `http://www.iwr.uni-heidelberg.de/groups/comopt/software`

[12] V. Chvátal (1973) Edmonds polytopes and a hierarchy of combinatorial problems. *Discr. Math.*, 4, 305–337.

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

[14] S. Cook (1971) The complexity of theorem proving procedures. *Proc. 3rd Ann. ACM Symp. Th. Comput. (STOC)*, pp. 151–158.

[15] G.B. Dantzig, D.R. Fulkerson & S.M. Johnson (1954) Solution of a large-scale traveling salesman problem. *Oper. Res.*, 2, 363–410.

[16] M. Deza, M. Grötschel & M. Laurent (1992) Clique-web facets for multicut polytopes. *Math. Oper. Res.*, 17, 981–1000.

[17] L.K. Fleischer, A.N. Letchford & A. Lodi (2006) Polynomial-time separation of a superclass of simple comb inequalities. *Math. Oper. Res.*, 31, 696–713.

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

[19] M.R. Garey & D.S. Johnson (1979) *Computers and Intractability: A Guide to the Theory of $\mathcal{NP}$-Completeness.* New York: Freeman.

[20] A.H.M. Gerards & A.J. Schrijver (1986) Matrices with the Edmonds–Johnson property. *Combinatorica*, 6, 365–379.

[21] M. Giandomenico & A.N. Letchford (2006) Exploring the relationship between max-cut and stable set relaxations. *Math. Program.*, 106, 159-175.

[22] M. Grötschel, L. Lovász & A.J. Schrijver (1988) Stable sets in graphs. Chapter 9 of *Geometric Algorithms in Combinatorial Optimization.* New York: Wiley.

[23] M. Grötschel, M. Jünger & G. Reinelt (1985) Facets of the linear ordering polytope. *Math. Program.*, 33, 43–60.

[24] M. Grötschel & Y. Wakabayashi (1990) Facets of the clique partitioning polytope. *Math. Program.*, 47, 367–387.

[25] M. Hartmann, M. Queyranne & Y. Wang (1999) On the Chvátal rank of certain inequalities. In G. Cornuéjols, R.E. Burkard & G.J. Woeginger (eds.) *Proceedings of IPCO VII*, pp. 218–233. Heidelberg: Springer.

[26] M. Jünger *et al.* (eds.) *50 Years of Integer Programming: 1958-2008.* Berlin: Springer.

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

[28] A.N. Letchford, G. Reinelt & D.O. Theis (2008) Odd minimum cut-sets and b-matchings revisited. *SIAM J. Discr. Math.*, 22, 1480–1487.

[29] L. Lovász & A.J. Schrijver (1991) Cones of matrices and set-functions and 0-1 optimization. *SIAM J. Optim.*, 1, 166–190.

[30] J.F. Marcotorchino (1981) *Aggregation of Similarities in Automatic Classification* (in French). Doctoral Thesis, Université Paris VI.

[31] R. Müller (1996) On the partial order polytope of a digraph. *Math. Program.*, 73, 31–49.

[32] R. Müller & A.S. Schulz (2002) Transitive packing: a unifying concept in combinatorial optimization. *SIAM J. Optim.*, 13, 335–367.

[33] D. Naddef (2002) Polyhedral theory and branch-and-cut algorithms for the TSP. In G. Gutin & A.P. Punnen (eds.) *The Traveling Salesman Problem and Its Variations*, pp. 29–114. Dortrecht: Kluwer.

[34] H. Nagamochi, T. Ono & T. Ibaraki (1994) Implementing an efficient minimum capacity cut algorithm. *Math. Program.* 67, 325–341.

[35] M.W. Padberg (1973) On the facial structure of set packing polyhedra. *Math. Program.*, 5, 199–215.

[36] M.W. Padberg & M.R. Rao (1982) Odd minimum cut-sets and b-matchings. *Math. Oper. Res.*, 7, 67–80.

[37] M. Padberg & G. Rinaldi (1991) A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM Rev.*, 33, 60–100.

[38] L. Trevisan, G.B. Sorkin, M. Sudan & D.P. Williamson (2000) Gadgets, approximation, and linear programming. *SIAM J. Comput.*, 29, 2074–2097.