

A smaller extended formulation for the odd cycle inequalities of the stable set polytope

Sven de Vries^a, Bernd Perscheid^{a,*}

^a*Operations Research, FB IV - Mathematics, Trier University, 54286 Trier, Germany*

Abstract

For sparse graphs, the odd cycle polytope can be used to compute useful bounds for the maximum stable set problem quickly. Yannakakis [9] introduced an extended formulation for the odd cycle inequalities of the stable set polytope in 1991, which provides a direct way to optimize over the odd cycle polytope in polynomial time, although there can exist exponentially many odd cycles in given graphs in general. We present another extended formulation for the odd cycle polytope that uses less variables and inequalities than Yannakakis' formulation. Moreover, we compare the running time of both formulations as relaxations of the maximum stable set problem in a computational study.

Keywords: Stable set, Odd cycle inequalities, Separation algorithm, Extended formulation, Linear programming

1. Introduction

Let $G = (V, E)$ be a simple, undirected graph and assume $V = \{1, \dots, n\}$. A *stable set* is a subset $N \subseteq V$ with $ij \notin E$ for every pair $i, j \in N$. The incidence vector of N is denoted by $x^N \in \{0, 1\}^n$, where $x_i^N = 1$ if and only if $i \in N$. The stable set polytope $\text{STAB}(G)$ is the convex hull of incidence vectors of stable sets in G .

*Corresponding author

Email addresses: devries@uni-trier.de (Sven de Vries),
perscheid@uni-trier.de (Bernd Perscheid)

The maximum stable set problem can be formulated as

$$\begin{aligned} \max \quad & \sum_{i \in V} x_i \\ \text{s.t.} \quad & x_i + x_j \leq 1 && \forall ij \in E \\ & x_i \in \{0, 1\} && \forall i \in V \end{aligned}$$

and is \mathcal{NP} -hard. A weak linear relaxation can be given by relaxing the integrality constraints, i.e. replacing $x_i \in \{0, 1\}$ by $x_i \in [0, 1]$ for every $i \in V$. For each isolated vertex i in G , its variable x_i is equal to 1 in every maximum stable set in G . Thus, without loss of generality we consider only graphs without isolated vertices. Then, for every $i \in V$ there exists an edge $ij \in E$ and the corresponding edge inequality $x_i + x_j \leq 1$ together with $x_j \geq 0$ implies already $x_i \leq 1$. This allows to omit the constraint $x_i \leq 1$ for all $i \in V$. We call the resulting polytope

$$P^E(G) := \{x \in \mathbb{R}^n : x \text{ satisfies (1) and (2)}\}$$

with

$$x_i \geq 0 \quad \forall i \in V \quad (\text{trivial inequalities}) \quad (1)$$

$$x_i + x_j \leq 1 \quad \forall ij \in E \quad (\text{edge inequalities}) \quad (2)$$

the *edge constrained stable set polytope*, which is also known as the fractional stable set polytope and denoted by $\text{FRAC}(G)$.

Some well-known valid inequalities for $\text{STAB}(G)$ are the *odd cycle inequalities*. They are defined as

$$\sum_{i \in C} x_i \leq \frac{|C| - 1}{2} \quad \forall \text{ odd cycles } C \quad (3)$$

and constitute an exponential class of stable set inequalities in general. For the special case that C is a *chordless* odd cycle in G , Padberg [8] shows how to lift the odd cycle inequality for C to obtain an inequality that induces a facet of $\text{STAB}(G)$. By intersecting *all* odd cycle inequalities with $P^E(G)$, we obtain the odd cycle polytope

$$P^{OC}(G) = \{x \in \mathbb{R}^n : x \text{ satisfies (1), (2), and (3)}\}.$$

Graphs with $P^{OC}(G) = \text{STAB}(G)$ are called *t*-perfect and were first studied by Chvátal [2]. Although not many graphs are *t*-perfect, the odd cycle polytope can be used to determine upper bounds for the maximum

stable set problem very quickly. Notice that for any graph G , we have $(\frac{1}{3}, \dots, \frac{1}{3}) \in P^{OC}(G)$ which violates every clique inequality

$$\sum_{i \in K} x_i \leq 1 \quad (4)$$

for cliques K in G of size at least 4. This indicates that the quality of the bounds provided by $P^{OC}(G)$ is better the sparser the graph is. In Section 4 we show how to optimize over $P^{OC}(G)$ with extended formulations, whose best one consists of $2n^2$ variables and just $4mn + 3n$ inequalities.

2. Separation of odd cycle inequalities

Given two polytopes P^X and P with $P^X \subseteq P$, the separation problem for P^X and a given vector $\bar{x} \in P$ is to either confirm $\bar{x} \in P^X$ or to give a valid inequality of P^X that is violated by \bar{x} .

Many separation problems can be solved by means of auxiliary graphs. For this purpose, we define the so-called categorical graph product. Applying it to G and a specific graph H , it can be used to solve the separation problem for the odd cycle inequalities of the stable set polytope.

Definition 2.1 (Hammack [7]). *The categorical product $G \times H$ of a graph $G = (V_G, E_G)$ and a graph $H = (V_H, E_H)$ is given by the vertex set $V_{G \times H} = V_G \times V_H$ and the edge set $E_{G \times H} = \{(u, i), (v, j)\} : uv \in E_G \text{ and } ij \in E_H\}$.*

For the separation of odd cycle inequalities assume $\bar{x} \in [0, 1]^n$ fulfills the edge inequalities (2), that is $\bar{x} \in P^E(G)$. We define weights $\bar{w}_{ij} := 1 - \bar{x}_i - \bar{x}_j$ for every edge ij in $G = (V, E)$, cf. Grötschel et al. [5, proof of Lemma 9.1.11], which are obviously nonnegative since $\bar{x} \in P^E(G)$ and hence $\bar{x}_i + \bar{x}_j \leq 1$. Let C be an odd cycle in G . The corresponding odd cycle inequality

$$\sum_{i \in C} \bar{x}_i \leq \frac{|C| - 1}{2}$$

of $P^{OC}(G)$ is equivalent to

$$1 \leq |C| - 2 \sum_{i \in C} \bar{x}_i = \sum_{ij \in E_C} (1 - \bar{x}_i - \bar{x}_j) = \sum_{ij \in E_C} \bar{w}_{ij} =: \bar{w}(E_C).$$

This leads to the following statement.

Corollary 2.2. *Let $\bar{x} \in P^E(G)$. All odd cycle inequalities are fulfilled by \bar{x} if and only if every odd cycle C in G has weight $\bar{w}(E_C)$ at least 1.*

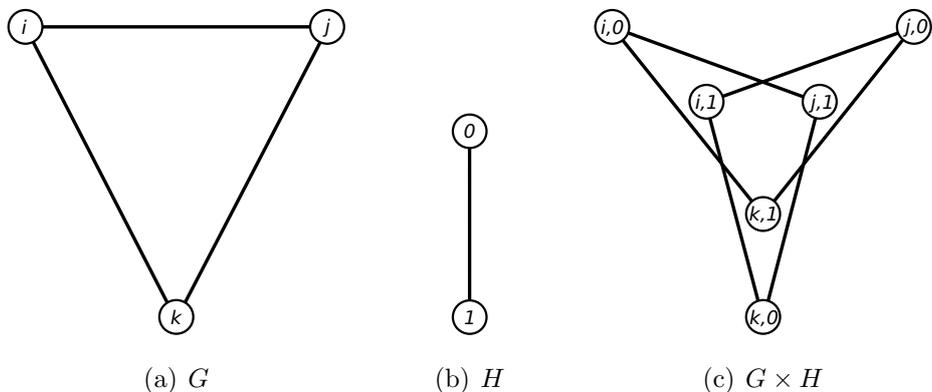


Figure 1: The categorical product of C_3 and an edge.

Accordingly, computing a shortest odd cycle in G with respect to edge weights \bar{w} suffices to solve the separation problem for the odd cycle inequalities. We describe how to find such a shortest odd cycle in G as presented by Grötschel et al. [5, Chapter 8.3]. Besides that, we interpret their construction through the concept of product graphs.

Consider the categorical product of $G = (V_G, E_G)$ and $H = (V_H, E_H)$ with $V_H = \{0, 1\}$ and $E_H = \{\{0, 1\}\}$. An example is given in Figure 1, where G is a 3-cycle.

For every edge ij in G , we assign the weight \bar{w}_{ij} to the edges $\{(i, 0), (j, 1)\}$ and $\{(i, 1), (j, 0)\}$ in $G \times H$. Since all edge weights \bar{w} are nonnegative, shortest paths in $G \times H$ can be computed efficiently, for instance with Dijkstra's algorithm [4]. Then the weight of a shortest odd cycle in G can be found in the following way: compute a shortest path between vertices $(i, 0)$ and $(i, 1)$ in H for every $i \in V_G$, then choose (one of) the shortest among these paths. The weight of this path is equal to the weight of a shortest odd cycle in G . Checking whether this weight is at least 1 solves the separation problem for the odd cycle inequalities.

Notice, a shortest path between a pair $(i, 0)$ and $(i, 1)$ of vertices in $G \times H$ corresponds to an odd closed walk in G , which can be an odd *cycle*, e.g. the 3-cycle (i, j, k, i) in G arises from the $(i, 0)$ - $(i, 1)$ -path in Figure 2(a). In general, this odd closed walk in G needs not necessarily be an odd cycle in G , even if we consider a shortest of all shortest $(i, 0)$ - $(i, 1)$ -paths for $i \in V_G$. For example, the $(i, 0)$ - $(i, 1)$ -path in Figure 2(b) yields the odd closed walk (i, j, k, l, j, i) in G , which is not a cycle. In this case, there exists some subpath between $(u, 0)$ and $(u, 1)$ in $G \times H$, whose weight is equal to the weight of a shortest $(i, 0)$ - $(i, 1)$ -path and which gives an odd cycle in G . It cannot have less weight, since this would contradict the minimality of the weight of

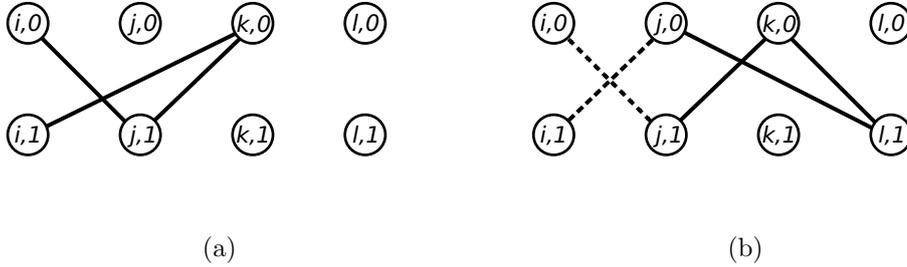


Figure 2: Shortest $(i, 0)$ - $(i, 1)$ -paths in $G \times H$.

the $(i, 0), (i, 1)$ -path. Moreover, the nonnegative edge weights ensure that its weight cannot exceed the weight of the $(i, 0)$ - $(i, 1)$ -path. In our example, this is a $(j, 0)$ - $(j, 1)$ -path, which corresponds to the odd cycle (j, k, l, j) in G .

3. Shortest path LPs

Let us recapitulate techniques presented by Ahuja et al. [1] to solve shortest path problems in digraphs without negative weight cycles via linear programs. Furthermore, we deduce a feasibility problem from one of these optimization problems. The arising polyhedron will be used to derive the extended formulation Q_0^{OC} in Section 4.

As we are given an undirected graph $G = (V, E)$, we initially transform it into a digraph $D = (V, A)$ by replacing every edge $ij \in E$ by two antiparallel arcs (i, j) and (j, i) . All these arcs constitute A . Notice that there are no negative weight cycles in D since all edge weights in G are nonnegative.

Let s and t be fixed vertices in V . We are able to find a shortest s - t -path by solving the minimum cost flow problem

$$\begin{aligned}
 \min \quad & \sum_{(i,j) \in A} w_{ij} x_{ij} \\
 \text{s.t.} \quad & \sum_{j: (i,j) \in A} x_{ij} - \sum_{j: (j,i) \in A} x_{ji} = \begin{cases} 1, & i = s \\ -1, & i = t \\ 0, & i \in V \setminus \{s, t\} \end{cases} \quad (\text{MCF-SP}) \\
 & x_{ij} \geq 0 \quad \forall (i, j) \in A.
 \end{aligned}$$

This linear program can be interpreted as sending one unit of flow from s to t through the network. The variables x_{ij} represent the traffic on arc (i, j) and obviously have to be nonnegative. The other constraints ensure that

one unit leaves the source s , one unit arrives at the sink t and that the flow conservation property is fulfilled. A solution of this minimum cost flow problem then gives the weight of a shortest s - t -path.

Using a dual relation as proposed by Ahuja et. al. [1], the optimal value of the linear program

$$\begin{aligned} \max \quad & y_{st} \\ \text{s.t.} \quad & y_{ss} = 0 \\ & y_{sj} \leq y_{si} + w_{ij} \quad \forall (i, j) \in A \end{aligned} \tag{DMCF-SP}$$

also gives the weight of a shortest s - t -path. Notice that strictly speaking the objective is to maximize $y_{st} - y_{ss}$ where $y_{ss} = 0$ due to the first constraint. In the case that no s - t -path exists, y_{st} is unbounded.

The following system of (in)equalities arises from (DMCF-SP) and has feasible points if and only if the weight of all s - t -paths is at least c :

$$\begin{aligned} y_{ss} &= 0 \\ y_{sj} &\leq y_{si} + w_{ij} \quad \forall (i, j) \in A \\ y_{st} &\geq c \end{aligned} \tag{F-SP}$$

4. Extended formulations for odd cycles

In this section, we present three different extended formulations that can be used to optimize over $P^{OC}(G)$ in polynomial time. The first one arises naturally from the separation procedure in Section 2 in combination with the ideas of the previous section. The second one is the well-known extended formulation of Yannakakis [9]. The third formulation is more efficient as it combines the strengths of the other two formulations.

4.1. A direct approach

A natural approach to obtain an extended formulation for the odd cycle polytope $P^{OC}(G)$ is to use the idea at the end of Section 3 and adapt it to every pair $(i, 0)$ and $(i, 1)$ with $i \in V$ in the product graph $G \times H$ from Section 2 where we replace every edge by two antiparallel arcs. We define $A_{G \times H} := \{((i, r), (j, s)) : \{(i, r), (j, s)\} \in E_{G \times H}\}$. For every edge ij in G , we have $\{(i, 0), (j, 1)\}$ and $\{(i, 1), (j, 0)\}$ in $E_{G \times H}$ and hence $|E_{G \times H}| = 2m$. Thus, $|A_{G \times H}| = 4m$. Since shortest $(i, 0)$ - $(i, 1)$ -paths in $G \times H$ have the same weight as shortest odd cycles through i , they should have weight at least 1, see Corollary 2.2. Thus, with $c = 1$ in (F-SP):

$$Q_0^{OC}(G) := \{(x, y) \in \mathbb{R}^n \times \mathbb{R}^{4n^2} : (x, y) \text{ satisfies (1), (2), and (5)–(7)}\}$$

with

$$y_{irir} = 0 \quad \forall i \in V, r \in \{0, 1\} \quad (5)$$

$$y_{irjs} \leq y_{irkt} + w_{ktjs} \quad \forall ((k, t), (j, s)) \in A_{G \times H}, i \in V, r \in \{0, 1\} \quad (6)$$

$$y_{i0i1} \geq 1 \quad \forall i \in V \quad (7)$$

is an extended formulation of $P^{OC}(G)$.

We compare the number of variables and inequalities of all extended formulations that are presented in this section. If we fix variables as constants, such as the y_{irir} in equations (5), we do not count them as variables and the respective equations are not counted as inequalities.

Remark 4.1. *The extended formulation $Q_0^{OC}(G)$ requires $n + 4n^2 - 2n = 4n^2 - n$ variables and has $n + m + 8mn + n = 8mn + m + 2n$ inequalities.*

4.2. Yannakakis' formulation

The following polyhedron is a well-known extended formulation of $P^{OC}(G)$, due to Yannakakis [9], which uses less variables and inequalities than $Q_0^{OC}(G)$. For easier counting of inequalities, we define $A := \{(i, j) : ij \in E\}$ and use it for inequalities (8)–(10) in contrast to the original formulation of Yannakakis (if, for example, variable f_{ij} is involved for edge $ij \in E$, then ij simultaneously produces an inequality for variable f_{ji}). Then $|A| = 2m$, since for every edge ij we have $(i, j) \in A$ and $(j, i) \in A$.

$$Q_1^{OC}(G) := \{(x, f, g) \in \mathbb{R}^n \times \mathbb{R}^{n^2} \times \mathbb{R}^{n^2} : (x, f, g) \text{ satisfies (1), (8)–(11)}\}$$

with

$$0 \leq f_{ij} \leq 1 - x_i - x_j \quad \forall (i, j) \in A \quad (8)$$

$$f_{ij} \leq f_{ik} + g_{kj} \quad \forall (i, k) \in A, j \in V \quad (9)$$

$$g_{ij} \leq f_{ik} + f_{kj} \quad \forall (i, k) \in A, j \in V \quad (10)$$

$$f_{ii} \geq 1 \quad \forall i \in V. \quad (11)$$

The edge inequalities (2) are implied by inequalities (8) in $Q_1^{OC}(G)$, because $0 \leq 1 - x_i - x_j$ is equivalent to $x_i + x_j \leq 1$ for all edges $ij \in E$. Inequalities (8), (9), and (10) imply that f_{ij} is bounded from above by the weight of a shortest *odd* walk between two vertices i and j , whereas g_{ij} is bounded from above by the weight of a shortest even walk using at least two edges between vertices i and j . Finally, inequalities (11) ensure that all odd cycle inequalities hold.

Remark 4.2. *Yannakakis' formulation $Q_1^{OC}(G)$ from above requires $2n^2 + n$ variables. Furthermore, $Q_1^{OC}(G)$ has $n+4m+2mn+2mn+n = 4mn+4m+2n$ inequalities.*

4.3. A smaller extended formulation

Analogously to $Q_1^{OC}(G)$, we add variables $f, g \in \mathbb{R}^{n^2}$ to the original variables $x \in \mathbb{R}^n$ for our new extended formulation $Q_2^{OC}(G)$ and define $A := \{(i, j) : ij \in E\}$. The set of (in)equalities of $Q_2^{OC}(G)$ combines the benefits of $Q_0^{OC}(G)$ and $Q_1^{OC}(G)$ with the idea that the variables that represent the weight of shortest even closed walks can be fixed to 0.

$$Q_2^{OC}(G) := \{(x, f, g) \in \mathbb{R}^n \times \mathbb{R}^{n^2} \times \mathbb{R}^{n^2} : (x, f, g) \text{ satisfies (1), (12)–(15)}\}$$

with

$$g_{ii} = 0 \quad \forall i \in V \quad (12)$$

$$f_{ij} \leq g_{ik} + 1 - x_k - x_j \quad \forall (k, j) \in A, i \in V \quad (13)$$

$$g_{ij} \leq f_{ik} + 1 - x_k - x_j \quad \forall (k, j) \in A, i \in V \quad (14)$$

$$f_{ii} \geq 1 \quad \forall i \in V. \quad (15)$$

Notice that the edge inequalities are automatically fulfilled in the formulation above. However, this is less obvious than in the formulation $Q_1^{OC}(G)$:

Lemma 4.3. *The edge inequalities (2) are implied by the set of inequalities of $Q_2^{OC}(G)$.*

Proof. Let $uv \in E$ be some edge for which we want to show that $x_u + x_v \leq 1$ holds. Consider inequalities (13) with $i = k = u$ and $j = v$. Then we have

$$f_{uv} \leq g_{uu} + 1 - x_u - x_v.$$

For inequalities (14), let $i = j = u$ and $k = v$. Thus,

$$g_{uu} \leq f_{uv} + 1 - x_v - x_u.$$

With $g_{uu} = 0$ by equations (12), adding both inequalities from above yields

$$f_{uv} \leq f_{uv} + 2(1 - x_u - x_v),$$

which reduces to

$$x_u + x_v \leq 1.$$

□

Theorem 4.4. $Q_2^{OC}(G)$ is an extended formulation of $P^{OC}(G)$.

Proof. Let $\bar{x} \in P^{OC}(G)$. We show that there exist \bar{f} and \bar{g} so that $(\bar{x}, \bar{f}, \bar{g}) \in Q_2^{OC}(G)$ holds. All inequalities (1) occur in both polytopes and they are not violated by \bar{x} . Assign the nonnegative weight $\bar{w}_{ij} = 1 - \bar{x}_i - \bar{x}_j$ to every edge $ij \in E$ and hence to every arc $(i, j) \in A$. Furthermore, define \bar{f}_{ij} for all $i, j \in V$ and \bar{g}_{ij} for all $i, j \in V$ as the weights of shortest odd and even walks, respectively, between vertices i and j in G (assign a large value to the corresponding variable if no such walk exists). Therefore, the variables f and g are symmetric, i.e. $\bar{f}_{ij} = \bar{f}_{ji}$ and $\bar{g}_{ij} = \bar{g}_{ji}$ for all $i, j \in V$. Notice that, because there are no cycles with negative weight, a shortest even walk for every $i \in V$ to itself can simply use zero edges and therefore has weight 0. This ensures that equations (12) hold. It is easy to see that inequalities (13) and (14) are not violated either due to the construction of \bar{f} and \bar{g} . These inequalities can be interpreted as follows: The weight of shortest odd or even walks from vertex i to j can not exceed the weight of a shortest walk of the opposite parity from i to k plus the weight of the arc (k, j) .

Inequalities (3) are satisfied by \bar{x} . This is, as mentioned above, equivalent to every odd cycle C having weight $\bar{w}(E_C)$ at least 1. Observe that \bar{f}_{ii} is the weight of a shortest odd closed walk starting in $i \in V$. If this walk is not a cycle, there always exists an odd cycle with weight less or equal to the weight of a shortest odd closed walk. However, this cycle does not necessarily include vertex i , but its weight has to be as well greater or equal to 1. Therefore $\bar{f}_{ii} \geq 1$ holds for every $i \in V$.

For the converse, let $(\bar{x}, \bar{f}, \bar{g}) \in Q_2^{OC}(G)$. No edge inequality is violated by \bar{x} , see Lemma 4.3. Inequalities (12), (13), and (14) ensure that \bar{f}_{ij} and \bar{g}_{ij} are bounded from above by the weights of shortest odd and even walks, respectively, between vertices i and j . Thus, the value \bar{f}_{ii} is lower or equal than the weight of a shortest odd cycle through vertex i . Since $\bar{f}_{ii} \geq 1$ for every $i \in V$ by inequalities (15), no odd cycle inequality is violated and hence $\bar{x} \in P^{OC}(G)$. \square

Remark 4.5. Formulation $Q_2^{OC}(G)$ requires just $n + 2n^2 - n = 2n^2$ variables and $n + 2mn + 2mn + n = 4mn + 2n$ inequalities.

4.4. Comparison

Both formulations $Q_1^{OC}(G)$ and $Q_2^{OC}(G)$ use the same variables f and g and they are extended formulations of $P^{OC}(G)$. One wonders about their relation: is one polyhedron a subset of the other one? Before we show why this is not the case, we point out one special property of the variables g_u in $Q_1^{OC}(G)$.

Lemma 4.6 (cf. de Vries et al. [3]). *Let $(\bar{x}, \bar{f}, \bar{g}) \in Q_1^{OC}(G)$ and $\hat{g}_{ij} := \bar{g}_{ij}$ for all $i, j \in V$ with $i \neq j$. Then for all \hat{g}_l in the interval $[0, 2 \min\{f_{lk} : (l, k) \in A\}]$ with $l \in V$ we have $(\bar{x}, \bar{f}, \hat{g}) \in Q_1^{OC}(G)$.*

Proof. For each $l \in V$ the variable g_l occurs in inequalities (9), i.e. $f_{il} \leq f_{il} + g_l$ if $j = k = l$. Therefore it is bounded from below by zero and there is no tighter lower bound given by the set of inequalities. On the other hand the variable g_l is bounded from above by inequalities (10) when $i = j = l$. In this case we obtain $g_l \leq f_{lk} + f_{kl}$ for every $(l, k) \in A$. We have $2 \min\{f_{lk}, f_{kl}\} \leq f_{lk} + f_{kl}$ for all $(l, k) \in A$ and there is no upper bound that is smaller than $f_{lk} + f_{kl}$. Therefore $(\bar{x}, \bar{f}, \hat{g}) \in Q_1^{OC}(G)$. \square

Theorem 4.7. *In general, $Q_1^{OC}(G) \not\subseteq Q_2^{OC}(G)$.*

Proof. Let w.l.o.g. $\bar{x} = \mathbf{0}$ and $\bar{f} = \mathbf{1}$. Then $\bar{g}_l = 2$ is feasible for all $l \in V$, since $\bar{g}_l = 2 \in [0, 2 \min\{f_{lk} : (l, k) \in A\}] = [0, 2]$, cf. Lemma 4.6. All the other variables \bar{g}_{ij} with $i \neq j$ can attain every value from the interval $[0, 2]$. It follows that $(\bar{x}, \bar{f}, \bar{g}) \in Q_1^{OC}(G)$. Moreover, $\bar{g}_l = 2$ for all $l \in V$ violates inequalities (12) and thus $(\bar{x}, \bar{f}, \bar{g}) \notin Q_2^{OC}(G)$. \square

Theorem 4.8. *In general, $Q_2^{OC}(G) \not\subseteq Q_1^{OC}(G)$.*

Proof. Consider $(\bar{x}, \bar{f}, \bar{g})$ with $\bar{x} = \mathbf{0}$, $\bar{g} = \mathbf{0}$, $\bar{f}_{ii} = 1$ for all $i \in V$, and $\bar{f}_{ij} = -1$ for all $i, j \in V$ with $i \neq j$. Then $(\bar{x}, \bar{f}, \bar{g}) \in Q_2^{OC}(G)$, but $(\bar{x}, \bar{f}, \bar{g}) \notin Q_1^{OC}(G)$, since inequalities (8) restrict variables f_{ij} to be non-negative if $(i, j) \in A$. \square

5. Numerical Results

In the previous section, three different extended formulations for the odd cycle polytope $P^{OC}(G)$ are presented. The polyhedron $Q_0^{OC}(G)$ has about twice as many variables as $Q_1^{OC}(G)$ and $Q_2^{OC}(G)$. Furthermore, its coefficient of the dominating term mn for the number of inequalities is twice that of the two other polyhedra. Therefore, we restrict our computational study to the running times of solving $Q_1^{OC}(G)$ and $Q_2^{OC}(G)$. We use CPLEX v. 12.8.0.0 as an LP solver for an extensive set of test instances. These instances are generated randomly with the graph generator *fast_gnp_random_graph* from the Python package NetworkX [6]. The input data are the number of vertices n and a probability p for each pair i and j of vertices in G to share an edge. We tabulate the graphs with respect to n and p (that equals the expected density d). For each pair of parameters n and d we consider, we compute the average CPLEX running time for optimizing over $Q_1^{OC}(G)$ and $Q_2^{OC}(G)$ on 100 different test instances.

Table 1: Average running time of $Q_1^{OC}(G)$ and $Q_2^{OC}(G)$ with the dual simplex method; each number is computed as the average over 100 instances.

n	d (%)	Dual simplex method				
		Average gap (%)		Average CPU (s)		$\rho(G)$
		$z^E(G)$	$z^{OC}(G)$	$Q_1^{OC}(G)$	$Q_2^{OC}(G)$	
50	5	0.83	0.03	1.11	1.06	5 %
	10	15.09	0.61	18.52	3.15	83 %
	15	29.25	4.28	162.23	4.64	97 %
	20	40.24	12.58	563.51	5.88	99 %

Let $\text{rt}(P^X(G))$ be the running time in seconds consumed by optimizing over the stable set relaxation $P^X(G)$. We define

$$\rho(G) := \frac{\text{rt}(Q_1^{OC}(G)) - \text{rt}(Q_2^{OC}(G))}{\text{rt}(Q_1^{OC}(G))} \times 100,$$

that displays how much time is proportionately saved by the relaxation $Q_2^{OC}(G)$ in comparison to the relaxation $Q_1^{OC}(G)$.

We solved all the problems for $n = 50$ with the dual simplex method, which is the default method in CPLEX, see Table 1. The longest running time among all test instances consumed by solving $Q_1^{OC}(G)$ is 1,152 seconds, whereas solving $Q_2^{OC}(G)$ never took more than 11 seconds. The barrier method turns out to be by far the best method available for solving our linear programs. Thus, we solved problems for $n \in \{50, 75, 100, 125, 150, 175, 200\}$ with the barrier method and use up to 32 threads for parallelization, see Table 2. For both methods a time limit of 3,600 seconds is used, which was only exceeded by three test instances with $n = 175$ when applying the barrier method to $Q_1^{OC}(G)$. Among all remaining test instances that were solved with the barrier method, the worst running time was 307 seconds for solving $Q_1^{OC}(G)$ and 212 seconds for solving $Q_2^{OC}(G)$, respectively.

As mentioned in Section 1, $x_i = \frac{1}{3}$ for all $i \in V$ is more likely feasible for $\text{STAB}(G)$ in sparse than in dense graphs. Therefore, the odd cycle polytope might be a better approximation of $\text{STAB}(G)$ in sparse graphs. For illustration, we report the optimality gap

$$\text{gap}(z) := \left| \frac{z^* - z}{z} \right| \times 100,$$

where z^* is the objective value of an optimal solution of the maximum stable set problem and z is the objective value of an optimal solution of some relaxation. For every set of test instances, the average gap of $z^E(G)$ and $z^{OC}(G)$,

Table 2: Average running time of $Q_1^{OC}(G)$ and $Q_2^{OC}(G)$ with the barrier method; each number is computed as the average over 100 instances.

n	d (%)	Barrier method					$\rho(G)$	
		Average gap (%)		Average CPU (s)		$Q_1^{OC}(G)$		$Q_2^{OC}(G)$
		$z^E(G)$	$z^{OC}(G)$					
50	5	0.83	0.03	1.04	1.12	-8 %		
	10	15.09	0.61	2.08	2.11	-1 %		
	15	29.25	4.28	2.99	2.58	14 %		
	20	40.24	12.58	3.82	3.48	9 %		
75	5	5.65	0.18	4.13	3.82	8 %		
	10	28.86	5.34	8.42	6.46	23 %		
	15	43.23	16.52	12.33	9.43	24 %		
	20	52.72	29.08	15.48	11.76	24 %		
100	5	14.35	0.97	12.08	9.67	20 %		
	10	38.82	13.41	22.95	16.05	30 %		
	15	51.96	28.04	33.29	22.33	33 %		
125	5	21.93	3.51	25.28	22.01	13 %		
	10	45.68	20.21	50.96	37.88	26 %		
	15	57.86	36.78	70.70	53.19	25 %		
150	5	28.52	7.74	53.64	35.57	34 %		
	10	50.95	26.75	97.90	59.84	39 %		
	15	62.27	43.40	146.67	88.88	39 %		
175	5	33.34	11.36	86.99	62.09	29 %		
	10 ¹	55.16	32.81	163.67	100.74	38 %		
200	5	38.02	15.22	147.53	86.08	42 %		
	10	58.84	38.28	275.46	139.82	49 %		

the objective values arising from optimizing over $P^E(G)$ and $P^{OC}(G)$, respectively, is given in Table 2.

Table 1 shows that the dual simplex method gets really slow on larger instances when optimizing over $Q_1^{OC}(G)$. The barrier method performs better than the dual simplex method on both formulations $Q_1^{OC}(G)$ and $Q_2^{OC}(G)$. However, optimizing over $Q_2^{OC}(G)$ is much faster than optimizing over $Q_1^{OC}(G)$.

6. Conclusion

Our new extended formulation allows for polynomial time optimization over the odd cycle polytope of the maximum stable set problem, although the number of odd cycles in a given graph is of exponential size in general. It improves Yannakakis' formulation from 1991, since it requires less inequalities. As our numerical experiments demonstrate, it can be solved faster in practice. We achieve average time savings of 24% on our set of test instances.

¹Three test instances were excluded, because the time limit of 3,600 seconds was reached while solving $Q_1^{OC}(G)$. Nevertheless, $Q_2^{OC}(G)$ was solved in less than 90 seconds for each of the excluded instances.

Acknowledgements

The authors were supported by the Research Training Group 2126 *Algorithmic Optimization* (ALOP), funded by the German Research Foundation DFG.

References

- [1] Ahuja, R. K., Magnanti, T. L., Orlin, J. B., 1993. *Network Flows: Theory, Applications and Algorithms*. Prentice-Hall.
- [2] Chvátal, V., 1975. On certain polytopes associated with graphs. *Journal of Combinatorial Theory, Series B* 18 (2), 138–154.
- [3] de Vries, S., Friedrich, U., Perscheid, B., 2020. An extended formulation for the 1-wheel inequalities of the stable set polytope. *Networks* 75 (1), 86–94.
- [4] Dijkstra, E. W., 1959. A note on two problems in connexion with graphs. *Numerische Mathematik* 1 (1), 269–271.
- [5] Grötschel, M., Lovász, L., Schrijver, A., 1988. *Geometric Algorithms and Combinatorial Optimization*. Springer.
- [6] Hagberg, A., Swart, P., Schult, D., 2008. Exploring network structure, dynamics, and function using networkx. Tech. rep., Los Alamos National Lab., Los Alamos, NM, USA.
- [7] Hammack, R., Imrich, W., Klavžar, S., 2011. *Handbook of Product Graphs*. CRC press.
- [8] Padberg, M. W., 1973. On the facial structure of set packing polyhedra. *Mathematical Programming* 5 (1), 199–215.
- [9] Yannakakis, M., 1991. Expressing combinatorial optimization problems by linear programs. *Journal of Computer and System Sciences* 43, 441–466.