Minimum clique cover in claw-free perfect graphs and the weak Edmonds-Johnson property

Flavia Bonomo[1], Gianpaolo Oriolo[2], Claudia Snels[2], and Gautier Stauffer[3]

[1] IMAS-CONICET and Departamento de Computación, FCEN, Universidad de Buenos Aires, Argentina. fbonomo@dc.uba.ar
[2] Dipartimento di Ingegneria Civile e Ingegneria Informatica, Università Tor Vergata, Roma, Italy. {oriolo,snels}@disp.uniroma2.it
[3] Bordeaux Institute of Mathematics, France. gautier.stauffer@math.u-bordeaux1.fr

Abstract. We give new algorithms for the minimum (weighted) clique cover in a claw-free perfect graph G, improving the complexity from $O(|V(G)|^5)$ to $O(|V(G)|^3)$. The new algorithms build upon neat reformulations of the problem: it basically reduces either to solving a 2-SAT instance (in the unweighted case) or to testing if a polyhedra associated with the edge-vertex incidence matrix of a bidirected graph has an integer solution (in the weighted case). The latter question was elegantly answered using neat polyhedral arguments by Schrijver in 1994. We give an alternative approach to this question combining pure combinatorial arguments (using techniques from 2-SAT and shortest paths) with polyhedral ones. Our approach is inspired by an algorithm from the Constraint Logic Programming community and we give as a side benefit a formal proof that the corresponding algorithm is correct (apparently answering an open question in this community). Interestingly, the systems we study have properties closely connected with the so-called Edmonds-Johnson property and we study some interesting related questions.

Keywords: clique cover,claw-free perfect graphs, bidirected graphs, Edmonds-Johnson property

1 Introduction

Given a graph G, a *clique cover* is a collection $\mathcal{K}$ of cliques covering all the vertices of G. Given a weight function $w : V(G) \mapsto \mathbb{Q}$ defined on the vertices of G, a *weighted clique cover* of G is a collection of cliques $\mathcal{K}$, with a positive weight y_K assigned to each clique K in the collection, such that, for each vertex v of G, $\sum_{K \in \mathcal{K}: v \in K} y_K \geq w(v)$. A *minimum clique cover* of G (MCC) is a clique cover of minimum cardinality, while a *minimum weighted clique cover* of G (MWCC) is a weighted clique cover minimizing $\sum_{K \in \mathcal{K}} y_K$.

For perfect graphs, it is well-known [5, 20] that the convex hull of the incidence vectors of all stable sets is described by clique inequalities and non-negativity constraints. It follows that the maximum weighted stable set (MWSS) problem (the left program) and the MWCC problem (the right program) form a primal-dual pair:

$$\max \sum_{v \in V} w(v) x_v \qquad\qquad \min \sum_{C \in \mathcal{K}(G)} y_C$$

$$\sum_{v \in C} x_v \leq 1 \quad \forall C \in \mathcal{K}(G) \qquad\qquad \sum_{C \in \mathcal{K}(G): v \in C} y_C \geq w(v) \quad \forall v \in V$$

$$x_v \geq 0 \quad \forall v \in V \qquad\qquad y_C \geq 0 \quad \forall C \in \mathcal{K}(G)$$

Moreover, when w is integral, there always exists an integer solution to the MWCC problem, as it was originally shown by Fulkerson [10].

In 1988, Grötschel, Lovász and Schrijver [12] gave a (non-combinatorial) polynomial time algorithm, building upon Lovász's theta function, to compute solutions to the MWSS problem and the MWCC problem in perfect graphs. It is a major open problem in combinatorial optimization whether there exist polynomial time combinatorial algorithms for those two problems.

For particular classes of perfect graphs, such algorithms exist. This is the case, for instance, for claw-free perfect graphs: a graph is *claw-free* if none of its vertices has a stable set of size three in its neighborhood. Claw-free graphs are a superclass of line graphs, and the MWSS problem in claw-free graphs is a generalization of the matching problem, and in fact there are several polynomial time combinatorial algorithms for solving the former problem (see [23]) and the fastest algorithm [9] runs in time $O(|V(G)|^2 \log |V(G)| + |V(G)||E(G)|)$. Conversely, to the best of our knowledge, the only combinatorial algorithm for the MWCC problem in the (entire) class of claw-free perfect graphs is due to Hsu and Nemhauser [15] in 1984 and runs in $O(|V(G)|^5)$. The algorithm is based on a clever use of complementary slackness in linear programming, combined with the resolution of several MWSS problems. Hsu and Nemhauser also designed a more efficient algorithm for the unweighted case [14], that runs in $O(|V(G)|^4)$. However, building *non-trivially* upon the clique cutset decomposition theorems for claw-free perfect graphs by Chvátal and Sbihi [6] and Maffray and Reed [19] and the algorithmic approach by Whitesides [27], one may design an $O(|V(G)|^3 \log |V(G)|)$-time algorithm for the MCC problem and a more involved $O(|V(G)|^4)$-time algorithm for the MWCC problem (for the latter result, one needs to use some ideas from [4], where an $O(|V(G)|^3)$-time algorithm for solving the MWCC problem on the subclass of *strip-composed* claw-free perfect graphs is given). [We defer the (long) details for this approach to the journal version of this paper.]

Our new approach to the problem relies on testing and building integer solution to systems of inequalities with at most 2 non-zero coefficients per row, both of them in $\{-1, +1\}$. We study in a slightly more general problem: given an $m \times n$ matrix A satisfying

$$\sum_{j=1}^{n} |a_{ij}| \leq 2, \text{ for all } i = 1, .., m, \text{ with } a_{ij} \in \mathbb{Z} \text{ for all i,j} \qquad (1)$$

(i.e., A is the vertex-edge incidence matrix of a bidirected graph, see Chapter 36 in [23] for more properties of those systems) and an integer vector b, can one determine in polynomial time if the system $Ax \leq b$ has an integer solution (and build one if any)? So we are interested in the polyhedron $P_b(A) := \{x \in \mathbb{R}^n : Ax \leq b\}$, and in particular in knowing if the integer hull of $P_b(A)$, that we denote by $Int(P_b(A))$, is empty or not: we sometimes refer to this question as the *integer feasibility* for $P_b(A)$. (When A is clear from the context, we abuse notation and denote $P_b(A)$ by P_b.) Note that all inequalities in $P_b(A)$ are of the type $x_i + x_j \leq b_{ij}$, $-x_i - x_j \leq b_{ij}$, $x_i - x_j \leq b_{ij}$, $x_i \leq b_i$, $-x_i \leq b_i$, $2x_i \leq b_i$, $-2x_i \leq b_i$.

We just pointed out that addressing efficiently the is question of integer feasibility for $P_b(A)$ leads to improved algorithm for MWCC in claw-free perfect graphs. However those systems are interesting for their own sake, as they also appear in other contexts, like for instance hardware and software verification [2], and, they received considerable attention from the Constraint Logic Programming community, as we recall later.

Schrijver [22] was, to the best of our knowledge, the first to consider this question (and he was motivated by some path problem in planar graphs!), and he gave an $O(n^3)$-time algorithm based on the Fourier-Motzkin elimination scheme that also produces a feasible integer solution when it exists.

An alternative to Schrijver's approach is that of Peis [21]. She reduces the problem of checking whether $Int(P_b) = \emptyset$ to, first, testing for fractional feasibility, i.e. if $P_b = \emptyset$, through shortest paths techniques. If P_b is non-empty, she gets a half integral solution certificate as a side benefit of the shortest path calculation. Then she tests if the fractional components of this half integral solution can be "rounded" up or down to an integer solution (solving a suitable 2-SAT problem). She proves that there always exists such a rounding procedure when a feasible integer solution exists. Like Schrijver's approach, her method is constructive, i.e. she builds a feasible integer solution when the system is integer non-empty. Her algorithm can be implemented to run in time $O(nm)$.

The result of Schrijver, and the more recent work of Peis, do not seem to be very well known, as several people in the Constraint Logic Programming community developed alternative algorithms and arguments for the problem (see e.g. [16, 13, 18, 25, 2, 24]), apparently ignoring the (previous) result in [22]. Interestingly though, the focus of this community is slightly different. They are not only interested in the integer feasibility, but they want to build efficiently what they call the *tight closure* of the system to possibly derive additional structural properties.

The best algorithm [24] to derive the tight closure runs in time $O(n^2 \log n + mn)$ (note that this is better than $O(n^3)$ as $m = O(n^2)$ when A satisfies (1)), while the best algorithm for testing integer feasibility runs in $O(nm)$ [18]. It seems however that all those results were pretty controversial in this community as they all rely on a fundamental theorem claimed in [16] that was never proved formally, as pointed out by [2] who declare in their paper "to present and, for the first time, fully justify an $O(n^3)$ algorithm to compute the tight closure of a set of UTVPI integer constraints".

We outline now the main contributions of each section. In Section 2, we discuss a new $O(|V(G)|^3)$-time, very simple, algorithm for the minimum (cardinality) clique cover in claw-free perfect graphs. We then extend our finding and devise a $O(V(G)|^3)$ algorithm for the weighted case thanks to Schrijver's result for matrices satisfying (1). In Section 3, we revisit from a polyhedral perspective the algorithm proposed in [24] for the integer feasibility and tight closure of systems $Ax \leq b$, with A satisfying (1), and offer a self-contained proof for its correctness and running time. We believe that this contribution is important as it bridges the gap between the CP community and the integer programming one, and also yields the tight closure (this is not possible neither with the approach of Schrijver, nor with that of Peis), and therefore addresses the different focus of the CP community. In Section 3.1, we introduce and study properties of those system that are closely related to the so-called Edmonds-Johnson property, and in Section 3.2 we identify a class of them with the following nice property: if the system has a fractional solution, then it has an integral one, and we show that this class includes the systems arising from the MWCC problem. [*For the sake of shortness some proofs will be postponed to the full version of this paper.*]

2 Clique covers in claw-free perfect graphs

We focus here on claw-free perfect graphs. We will give new $O(|V(G)|^3)$-time algorithms for the MCC and the MWCC problem. In particular, we will show how to "reduce" the latter problem to testing the existence of integer solution in polyhedra associated with the edge-incidence matrix of bidirected graphs. We start with the unweighted case.

2.1 A new algorithm for MCC in claw-free perfect graphs

Suppose that we are given a stable set S of a claw-free perfect graph $G = (V, E)$. We want to check if S is a maximum stable set of G. In the case that it is, we want to build a suitable clique cover of G of size $|S|$; in case it is not, we want to find an augmenting path (given a stable set S of a graph G, a path P is S–*alternating* if $(V(P) \setminus S) \cup (S \setminus V(P))$ is a stable set of G; S–*augmenting*, if in addition this stable set has size $|S| + 1$. Berge [3] proved that a stable set S is maximum for a claw-free graph G if and only if there are no paths that are S–augmenting). Without loss of generality we assume that S is maximal; therefore a vertex $v \in V \setminus S$ is either *bound*, i.e., it is adjacent to two vertices $s_1(v)$ and $s_2(v)$ of S, or is *free*, i.e., it is adjacent to one vertex $s(v)$ of S.

We will achieve our target by solving a suitable instance of the 2-SAT problem. The rationale is the following. By complementary slackness, in a perfect graph, every clique of a MCC intersects every MSS. Therefore, given S, in order to build a MCC we must "assign" each vertex of $v \in V \setminus S$ to a vertex in $N(v) \cap S$, in such a way that the set of vertices of $V \setminus S$ assigned to a same $s \in S$ will form a clique. As we show in the following, this can be easily expressed as a 2-SAT formula.

For every bound (resp. free) vertex $v \in V \setminus S$, we define two (resp. one) variables, or terms, $x_{vs_1(v)}$ and $x_{vs_2(v)}$ (resp. $x_{vs(v)}$) that will specify the above assignment. We also introduce an auxiliary boolean variable y to express that, for a free vertex v, $x_{vs(v)}$ has to be *true*. We consider three classes of clauses (we again denote by $\neg x_{vs}$ the negation of a term x_{vs}):

(c1) for each $v \in V \setminus S$ that is bound, $x_{vs_1(v)} \vee x_{vs_2(v)}$ must be true;
(c2) for each $s \in S$ and each $u, v \in N(s)$ that are non-adjacent, $\neg x_{us} \vee \neg x_{vs}$ must be true;
(c3) for each $v \in V \setminus S$ that is free, both $x_{vs(v)} \vee y$ and $x_{vs(v)} \vee \neg y$ must be true (i.e., $x_{vs(v)}$ must be true).

Consider the 2-SAT instance made of the conjunction of all the above clauses, which we denote in the following by the pair (G, S). It is straightforward to check that a clique cover of size $|S|$ induces a solution (i.e. a satisfying truth assignment) to (G, S). Vice versa, from a solution to (G, S) we can easily build a clique cover of size $|S|$ of G. In fact, for each vertex $s \in S$, let $X(s) := \{s\} \cup \{v \in N(s) : x_{vs} \text{ true}\}$. Note that for each free vertex u, following (c3), $u \in X(s(u))$. Moreover, for each $s \in S$, $X(s)$ is a clique, following (c2). Finally, following the clauses (c1), each bound vertex u belongs to either $X(s_1(u))$ or to $X(s_2(u))$. The family $\{X(s), s \in S\}$ is then a clique cover of size $|S|$. Therefore, a maximal stable set S is a maximum stable set of G if and only if there exists a solution to the 2-SAT instance (G, S). Moreover, from a solution to (G, S) we can easily build a MCC of G.

Following the above discussion, in order to design an algorithm for the MCC problem of a claw-free perfect graph G, we are left with the following question: what if S is *not* a maximum stable set of G, i.e. there is no solution to the 2-SAT instance (G, S)? In this case, in time $O|V(G)^2|$ we can find a path that is augmenting with respect to S. While we postpone the proof of this argument, that is rather standard, to the full version of the paper, we point that the search for this augmenting path is *not* technical, as we simply get it from a careful analysis of the implication graph of the unsatisfiable 2-SAT instance.

One therefore gets a simple algorithm that produces both a MCC and a MSS of a claw-free perfect graph G in time $O(|V(G)|^3)$, in the spirit of the augmenting path algorithm for maximum bipartite matching and minimum vertex cover.

2.2 A new algorithm for MWCC in claw-free perfect graphs

We are now given a claw-free perfect graph $G = (V, E)$ and also a weight function $w : V(G) \mapsto \mathbb{N} \setminus \{0\}$. Since w is strictly positive, every MWSS is maximal. We want to check if a given maximal stable set S of G is also a MWSS.

We will follow an approach inspired by the unweighted case. In that case, as in a perfect graph every clique of a MCC intersects every MSS, we tried to "assign" each vertex $v \in V \setminus S$ to a vertex in $N(v) \cap S$, so that the vertices of $V \setminus S$ assigned to a same $s \in S$ form a clique. In the weighted case, the assignment is no longer possible, as some vertices might have to be covered by several cliques in a MWCC. However, for each $v \in V \setminus S$ and $s \in N(v) \cap S$, we will *compute how much of $w(v)$ is covered by cliques that contain both s and v*. Therefore, for every bound (resp. free) vertex $v \in V \setminus S$, we define two (resp. one) non-negative integer variables $x_{vs_1(v)}$ and $x_{vs_2(v)}$ (resp. $x_{vs(v)}$), that will provide that information. We then consider the following constraints (note that, for $s \in S$ and $v \in N(s)$, x_{vs} is equivalent to either $x_{vs(v)}$, or $x_{vs_1(v)}$, or $x_{vs_2(v)}$):

(d1) for each $v \in V \setminus S$ that is bound, $x_{vs_1(v)} + x_{vs_2(v)} \geq w(v)$;
(d2) for each $v \in V \setminus S$ that is free: $x_{vs(v)} \geq w(v)$.
(d3) for each $s \in S$ and each $u, v \in N(s)$ that are non-adjacent, $x_{us} + x_{vs} \leq w(s)$;
(d4) for each $s \in S$ and each $u \in N(s)$, $x_{us} \leq w(s)$.

Consider the integer program P_b defined by the previous constraints, together with non-negativity and integrality for each variable. We claim that P_b has a (integer) solution if and only if there exists for G a (integer) weighted clique cover $(\mathcal{K}, y)$ with weight $w(S)$, i.e. if and only if S is a MWSS of G. Suppose there exists a weighted clique cover $(\mathcal{K}, y)$ of G with weight $w(S)$. Then S is a MWSS of G. It is straightforward to check that y induces a solution to P_b by letting, for each $s \in S$ and $v \in N(s)$, $x_{vs} = \sum_{K \in \mathcal{K}: s, v \in K} y_K$. Vice versa, let x be a (integer) solution to P. We want to "translate" x into a weighted clique cover $(\mathcal{K}, y)$ of weight $w(S)$. Let $s \in S$: we first take care of the weights of the cliques in the cover that contain s. So consider the graph $G^s = G[N[s]]$, with a weight function w^s defined as follows: for each vertex $v \in N(s)$, $w^s(v) = x_{vs}$; $w^s(s) = w(s)$. Trivially, $\{s\}$ is a MWSS of G^s, with respect to the weight function w^s, following constraints (d3)-(d4). Moreover, as every clique of G^s is a clique of G too, and $w^s(s) = w(s)$, if we compute a MWCC $(\mathcal{K}^s, y^s)$ of G^s (with respect to w^s), then the following holds: (j) for each vertex $v \in N(s)$, $\sum_{K \in \mathcal{K}^s: s, v \in K} y^s_K \geq x_{vs}$; (jj) $\sum_{K \in \mathcal{K}^s} y^s_K = w(s)$. Following constraints (d1)-(d2), if we compute, for *each* $s \in S$, a MWCC $(\mathcal{K}^s, y^s)$ of G^s, with respect to w^s, and we take $\mathcal{K} = \bigcup_{s \in S} \mathcal{K}^s$ and juxtapose the different y^s, $s \in S$, we then get a weighted clique cover $(\mathcal{K}, y)$ of G of weight $w(S)$.

We are left with two questions. The first, and more challenging one, is that of showing how it is possible to find an integral solution x to the system P_b defined by constraints (d1)-(d4). Observe that any inequality in (d1)-(d-4) involves at most two non-zero coefficients in $\{-1, +1\}$. Building integer solution to such systems can be done in $O(n^3)$ by an algorithm of Schrijver [22]. The second one is that of finding a MWCC of G^s with respect to the weight function w^s; we postpone to the full version of the paper the details, but this can be done in $O(|V(G^s)|^2)$-time. The overall complexity of this "translation" step is then $O(\sum_{s \in S} |V(G^s)|^2)$. By simple algebra, $\sum_{s \in S} |V(G^s)|^2 \leq (\sum_{s \in S} |V(G^s)|)^2$. But each $v \in V(G)$ belongs to at most two different graphs G^s, so $\sum_{s \in S} |V(G^s)| \leq 2|V(G)|$.

Our algorithm for the MWCC problem is summarized in the following: first compute a MWSS S of G, and then build a MWCC, as to run in $O(|V(G)|^3)$-time.

The MWSS S can be computed in $O(|V(G)|^3)$-time (cfr. [9]). A non-negative, integer solution x to P_b defined by constraints (d1)-(d4) can be found in $O(|V(G)|^3)$-time, see the next section and Section 3.3. Note also that, differently from the unweighted case, this algorithm does not use augmenting paths techniques to build *concurrently* a MWSS and a MWCC: we do not push this augmenting paths approach, as it would result in a $O(|V(G)|^4)$-time algorithm (we defer the details to the journal version).

3 $Ax \leq b$ when A satisfies (1), and b is integer

We are interested in the following problem: given an $m \times n$ matrix A satisfying (1) and an integer vector b, can one determine in polynomial time if the system $Ax \leq b$ has an integer solution (and build one if any)?

We associate to P_b (recall $P_b := \{x \in \mathbb{R}^n : Ax \leq b\}$) another polyhedron $Q_b \subseteq \mathbb{R}^{2n} := \{A' \begin{pmatrix} y \\ \bar{y} \end{pmatrix} \leq b'\}$ by associating inequalities to each inequality in the system $Ax \leq b$ as follows:

$$x_i + x_j \leq b_{ij} \rightarrow \begin{cases} y_i - \bar{y}_j \leq b_{ij} \\ -\bar{y}_i + y_j \leq b_{ij} \end{cases} \quad x_i \leq b_i \rightarrow y_i - \bar{y}_i \leq 2b_i$$

$$-x_i - x_j \leq b_{ij} \rightarrow \begin{cases} \bar{y}_i - y_j \leq b_{ij} \\ -y_i + \bar{y}_j \leq b_{ij} \end{cases} \quad 2x_i \leq b_i \rightarrow y_i - \bar{y}_i \leq b_i$$

$$-x_i \leq b_i \rightarrow -y_i + \bar{y}_i \leq 2b_i$$

$$x_i - x_j \leq b_{ij} \rightarrow \begin{cases} y_i - y_j \leq b_{ij} \\ -\bar{y}_i + \bar{y}_j \leq b_{ij} \end{cases} \quad -2x_i \leq b_i \rightarrow -y_i + \bar{y}_i \leq b_i$$

Lemma 1. *P_b has a solution if and only if Q_b has a solution.*

Proof. Necessity. Given a feasible solution $x^* \in P_b$, $(y^*, \bar{y}^*) : y_i^* = x_i^*, \bar{y}_i^* = -x_i^*$ for all i, is a solution to Q_b. *Sufficiency.* Given a feasible solution $(y^*, \bar{y}^*) \in Q_b$, $x^* : x_i^* = \frac{1}{2}y_i^* - \frac{1}{2}\bar{y}_i^*$ is a solution to P_b. We check it for the first type of inequality (i.e. to prove $x_i^* + x_j^* \leq b_{ij}$) but the method is the same for all 5 cases. If the inequality $x_i + x_j \leq b_{ij}$ is in the system $Ax \leq b$, by definition we have $y_i - \bar{y}_j \leq b_{ij}$ and $-\bar{y}_i + y_j \leq b_{ij}$ in the system defining Q_b. Taking the combination of those last two inequalities with multipliers $\frac{1}{2}$, $\frac{1}{2}$ yields $x_i^* + x_j^* \leq b_{ij}$. □

Observe that $(A')^t$ is a network matrix; we call D the corresponding (directed) graph, with cost b on its arcs. Any solution to Q_b defines what is usually called a feasible *potential* for D, and it follows from standard LP duality arguments that there is such a solution if and only if there are no negative cost cycles in D. In fact, given D, we can find in $O(nm)$-time either a feasible potential (integer potential as b is integer) or a negative cost cycle (see e.g. Theorem 7.7 in [17]).

For what follows, suppose therefore that Q_b has a feasible potential, i.e. $P_b \neq \emptyset$. The following lemma links the projection of P_b on each variable x_i, that we denote by $Proj_{x_i}(P_b)$, to the length of some suitable shortest paths in D, that e.g. can be computed in $O(mn + n^2 \log n)$-time by the algorithm of Moore-Bellman-Ford (see e.g. [17]).

Lemma 2. *If* $P_b \neq \emptyset$*, then* $Proj_{x_i}(P_b) = [\frac{p_i}{2}, \frac{q_i}{2}]$*, with* q_i *being the length of a shortest path from* $\bar{y}_i$ *to* y_i *in* D *(if any, else* $q_i = \infty$*), and* $-p_i$ *that of one from* y_i *to* $\bar{y}_i$ *(if any, else* $-p_i = \infty$*).*

Observe that if $\frac{p_i}{2}$ or $\frac{q_i}{2}$ are not integer values, then $x_i \geq \lceil \frac{p_i}{2} \rceil$ and $x_i \leq \lfloor \frac{q_i}{2} \rfloor$ are valid inequalities for the integer hull of P_b (recall that it is denoted by $Int(P_b)$). Therefore, if we are interested in the integer feasibility of P_b, i.e. if $Int(P_b)$ is empty or not, we can add those inequalities and define a new polyhedron $\overline{P}_b := P_b \cap \{x \in \mathbb{R}^n : \lceil \frac{p_i}{2} \rceil \leq x_i \leq \lfloor \frac{q_i}{2} \rfloor, i = 1, ..., n\}$ that is a tighter formulation for $Int(P_b)$.

Lemma 3. *Suppose that* $P_b \neq \emptyset$*. If, for each* i*,* $\lceil \frac{p_i}{2} \rceil \leq \lfloor \frac{q_i}{2} \rfloor$*, then* $Int(\overline{P}_b) = Int(P_b) \neq \emptyset$ *and we may find an integer solution to* P_b *in time* $O(n^3)$*.*

We would like to point out here that a slightly weaker result is implicit in Schrijver's approach (we defer the proof to the journal version of the paper).

Lemma 4. $Int(P_b) \neq \emptyset$ *if and only if, for each* i*,* $Proj_{x_i}(P_b)$ *has an integer point.*

The result is weaker than Lemma 3 in the sense that it does not tell us that the projection can be computed efficiently through shortest path (and actually Schrijver's approach does not even compute the projections). The next corollary follows from both lemmas (if we define $\overline{P}_b := P_b \cap \{x : \lceil \min_{x \in P_b} x_i \rceil \leq x_i \leq \lfloor \max_{x \in P_b} x_i \rfloor, \forall i = 1, ...n\}$ when using Lemma 4).

Corollary 1. $\overline{P}_b = \emptyset$ *if and only if* $Int(\overline{P}_b) = Int(P_b) \neq \emptyset$*.*

We close this section by linking with the results from the Constraint Logic Programming community. Because we can compute the transitive closure by shortest path calculation in D (this is immediate by definition of the transitive closure), our result also shows that we can compute the tight closure in time $O(n^2 logn + nm)$ (we apply the shortest path calculation twice). This is essentially the approach proposed in [24].

3.1 A weak Edmonds-Johnson property for matrices A satisfying (1)

Given a polyhedron $P = \{x \in \mathbb{R}^n : Ax \leq b\}$, we denote by P' its Chvátal-Gomory closure (or CG-closure), that is, the polytope obtained by adding to the system $Ax \leq b$ all its Chvatál-Gomory cuts (i.e., inequalities of the form $cx \leq \lfloor \delta \rfloor$, where c is an integer vector and $cx \leq \delta$ holds for each point in P.

A rational matrix A has the *Edmonds-Johnson property* if, for all d_1, d_2, b_1, b_2 integer vectors, the integer hull of

$$P = \{x \in \mathbb{R}^n : d_1 \leq x \leq d_2, b_1 \leq Ax \leq b_2\} \quad (2)$$

is given by P'. Edmonds and Johnson [7, 8] proved that if $A = (a_{ij})$ is an integral $m \times n$-matrix such that $\sum_{i=1}^{m} |a_{ij}| \leq 2$ for all $j = 1, .., n$, then A has the Edmonds-Johnson property. As shown by Gerards and Schrijver [11], the property does not hold when passing to transpose i.e. when A satisfies (1) as illustrated by taking A to be the edge-vertex incidence matrix of K_4 and then considering the system $0 \leq x \leq 1, 0 \leq Ax \leq 1$ (note that this is the linear relaxation of the edge formulation of the stable set polytope of K_4). Indeed it is easily proved that two rounds of Chvátal-Gomory cuts are needed in this case (one to produce all triangle inequalities, and one to produce the facet $x(V(K_4)) \leq 1$). In some sense, Gerards and Schrijver [11] prove that the converse holds i.e. matrix A satisfying (1) has the Edmonds-Johnson property if and only if it is the edge-vertex incidence matrix of a bidirected graph with no odd K_4-subdivision (see [11] for a proper definition). Moreover, in this case, optimizing over the integer hull of system (2) is easy, by the ellipsoid method, see [11] for more details; note that, if we only assume condition (1), there is no hope (unless $P = NP$) to optimize in polynomial time over the integer hull of (2), as one may encode the stable set problem.

We here define a *weaker* notion of Edmonds-Jonson property, that is mainly concerned with integer feasibility (recall that P' denotes the CG-closure of P):

Definition 1. *A rational matrix A has the* weak *Edmonds-Johnson property if, for all integer vectors d_1, d_2, b_1, b_2, the polyhedron $P = \{x \in \mathbb{R}^n : d_1 \leq x \leq d_2, b_1 \leq Ax \leq b_2\}$ has an integer solution if and only if P' is non-empty.*

By definition, the Edmonds-Johnson property implies the *weak* Edmonds-Johnson one, but the converse is not true. For instance, the edge-vertex incidence matrix of K_4 does not have the Edmonds-Johnson property but it has the weak Edmonds-Johnson one. In fact, we show in the following, *every* matrix A satisfying (1) has the property.

Theorem 1. *Every integral matrix B such that, for each i, $\sum_j |\bar{b}_{ij}| \leq 2$ has the weak Edmonds-Johnson property.*

Proof. Let A be a matrix satisfying (1). For each integer vector b, consider the polyhedron $P_b = \{x \in \mathbb{R}^n : Ax \leq b\}$. Without loss of generality, assume that $P_b \neq \emptyset$. We know from Corollary 1 that $Int(P_b) \neq \emptyset$ if and only if $\overline{P}_b \neq \emptyset$, where $\overline{P}_b := P_b \cap \{x : \left\lceil \min_{x \in P_b} x_i \right\rceil \leq x_i \leq \left\lfloor \max_{x \in P_b} x_i \right\rfloor, \forall i = 1, ...n\}$. Observe that $(P_b)' \subseteq \overline{P}_b$, as the inequalities $\lceil \min_{x \in P_b} x_i \rceil \leq x_i \leq \lfloor \max_{x \in P_b} x_i \rfloor$, are special CG-cuts for P_b. Therefore, $IP_b \neq \emptyset$ if and only if $(P_b)' \neq \emptyset$. The statement follows by observing that $\{x \in \mathbb{R}^n : d_1 \leq x \leq d_2, b_1 \leq Bx \leq b_2\}$ can be rewritten as $\{x \in \mathbb{R}^n : Ax \leq b\}$, with $b = (b_2, -b_1, d_2, -d_1)^t$, and $A = (B, -B, I, -I)^t$ satisfying (1). □

3.2 When CG-cuts are not needed

We would like to understand now under which conditions we do not need to add Chvátal-Gomory inequalities to P_b to ensure that fractional feasibility implies integer feasibility.

Observe that in the proof of Lemma 1, we retrieve a solution x of P_b from a solution $(y, \bar{y}) \in Q_b$ by taking a simple convex combination of the values y_i and $-\bar{y}_i$ (with multipliers $\frac{1}{2}$). We could try to see if other "convex combinations" yield valid solutions. For this purpose, we define $\Pi_A = \{\lambda \in [0,1]^q : A_2\lambda \leq \frac{A_2\mathbf{1}}{2}\}$, where A_2 is the submatrix of A made of those rows with $\sum_j |a_{ij}| = 2$. Observe that $\frac{A_2\mathbf{1}}{2} \in \{0, 1, -1\}^q$ and by definition $\lambda = \frac{\mathbf{1}}{2}$ is a feasible solution to Π_A. The system Π_A is made of inequalities of the type $\lambda_i - \lambda_j \leq 0, \lambda_i + \lambda_j \leq 1$, $-\lambda_i - \lambda_j \leq -1$, $2\lambda_i \leq 1$ and $-2\lambda_i \leq -1$. If we are interested in integer solution of Π_A, the last two restrictions impose $\lambda_i = 0$ and $\lambda_i = 1$ respectively. We call $\overline{\Pi_A}$ the polyhedra obtained from Π_A by substituting the restrictions $2\lambda_i \leq 1$ and $-2\lambda_i \leq -1$ with $\lambda_i = 0$ and $\lambda_i = 1$ respectively. All inequalities in $\overline{\Pi_A}$ can be rewritten under the form $\lambda_i + (1 - \lambda_j) \leq 1, \lambda_i + \lambda_j \leq 1$, $(1 - \lambda_i) + (1 - \lambda_j) \leq 1$, $\lambda_i \leq 0$ or $-\lambda_i \leq -1$ and thus $\overline{\Pi_A}$ can be trivially identified with the linear relaxation associated with the standard integer programming formulation of a 2-SAT instance. We have therefore:

Lemma 5. *Π_A has an integer solution if and only if the corresponding 2-SAT instance is satisfiable.*

The latter claim has the following nice consequence.

Lemma 6. *If Π_A has an integer solution, then P_b has an integer solution if and only if P_b is non-empty.*

The proof of Lemma 6 (that we postpone to the full version of the paper) shows that, when Q_b is non-empty, and we are given an integer solution λ to Π_A, one may always build an integer solution to P_b by (essentially) solving a single shortest path calculation. We sum-up the results obtained in the following:

Theorem 2. *If one knows a priori that Π_A has an integer solution, one can build an integer solution to P_b by solving a single shortest path problem and a single 2-SAT instance.*

Observe that any matrix A which is TU has the property that Π_A has an integer solution. This follows from the fact that A_2 is a submatrix of A and it is thus also TU, and that Π_A has the fractional solution $\frac{\mathbf{1}}{2}$. Interestingly, there are other 0,+/-1 matrices, that are not TU, that satisfy this property. For instance the matrix $A = \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}$. In general though, the fact that P_b has a integer solution does not imply that Π_A has one (consider for instance the system defined by the relations $x_1 + x_2 = 2, x_2 + x_3 = 2, x_3 + x_1 = 2$). However if we ask the property for all vector b and all subsystems (in the spirit of the definition of TU matrices), the converse holds, i.e. π_A has an integer solution, as Π_A is a special subsystem of P_b with $b = \frac{A\mathbf{1}}{2}$. We are currently investigating a proper definition of this kind to extend total unimodularity to the integer feasibility question, as we did for the weak Edmonds-Johnson property. We defer this to the journal version of the paper.

3.3 Back to minimum weighted clique cover

In the previous section we identified a class of systems that have a fractional solution if and only if they have an integral one. We now show that this class includes the systems arising from the MWCC problem.

We therefore go back to the algorithm in Section 2.2. So let S be a MWSS of a claw-free perfect graph G. We want to compute a non-negative, integer solution x to the system P_b defined by constraints $(d1)$-$(d4)$. Now let us give a look at the corresponding Π_A. Because we only keep those rows with two non-zero elements per row, Π_A reads:

$$\begin{aligned}\lambda_{us} + \lambda_{vs} &\leq 1, \forall s \in S, u, v \in N(s), uv \notin E \\ \lambda_{vs} + \lambda_{vs'} &\geq 1, \forall v \text{ bound, where } s, s' \text{ are the vertices in } S \cap N(v) \\ \lambda &\in [0,1]^q\end{aligned}$$

Now if there exists an integer solution to this system, there exists one with $\lambda_{vs} = 0$ for all v free (those vertices are only involved in the first type of constraints). Thus, integer feasibility for Π_A reduces to the existence of integer solutions to:

$$\begin{aligned}\lambda_{us} + \lambda_{vs} &\leq 1, \forall s \in S, u, v \in N(s), uv \notin E, u, v \text{ bound} \\ \lambda_{vs} + \lambda_{vs'} &\geq 1, \forall v \text{ bound, where } s, s' \text{ are the vertices in } S \cap N(v) \\ \lambda &\in [0,1]^n\end{aligned}$$

Note that this latter system has an integral solution if and only if there exists a clique cover of size $|S|$ in the graph $G[V \setminus F]$, where F is the set of the vertices that are free with respect to S. But this is trivially the case, as in $G[V \setminus F]$ there are no free vertices, and therefore no augmenting paths.

References

1. Aspvall, B., Plass, M., Tarjan, R.: A linear-time algorithm for testing the truth of certain quantified boolean formulas. Inf. Process. Lett. **8**(3) (1979) 121–123
2. Bagnara, R., Hill, P., Zaffanella, E.: An improved tight closure algorithm for integer octagonal constraints. In: Proc. 9th International Conference on Verification, Model Checking, and Abstract Interpretation. (2008) 8–21
3. Berge, C.: Graphs and Hypergraphs. Dunod, Paris (1973)
4. Bonomo, F., Oriolo, G., Snels, C.: Minimum weighted clique cover on strip-composed perfect graphs. In Golumbic, M., Stern, M., eds.: Proc. 38th WG. Volume 7551 of Lect. Notes Comput. Sci. (2012) 22–33
5. Chvátal, V.: On certain polytopes associated with graphs. J. Combin. Theory, Ser. B **18**(2) (1975) 138–154
6. Chvátal, V., Sbihi, N.: Recognizing claw-free perfect graphs. J. Combin. Theory, Ser. B **44** (1988) 154–176
7. Edmonds, J., Johnson, E.: Matching: a well-solved class of integer linear programs. In Guy, R., Hanani, H., Sauer, N., Schönheim, J., eds.: Combinatorial Structures and Their Applications. Gordon and Breach, New York (1970) 89–92

8. Edmonds, J., Johnson, E.: Matching, Euler tours and the Chinese postman. Math. Program. **5** (1973) 88–124
9. Faenza, Y., Oriolo, G., Stauffer, G.: An algorithmic decomposition of claw-free graphs leading to an $O(n^3)$-algorithm for the weighted stable set problem. In Randall, D., ed.: Proc. 22nd SODA, San Francisco, CA (2011) 630–646
10. Fulkerson, D.: On the perfect graph theorem. In Hu, T., Robinson, S., eds.: Mathematical Programming. Academic Press, New York (1973) 69–76
11. Gerards, A., Schrijver, A.: Matrices with the Edmonds-Johnson property. Combinatorica **6**(4) (1986) 365–379
12. Grötschel, M., Lovász, L., Schrijver, A.: Geometric Algorithms and Combinatorial Optimization. Springer, Berlin (1988)
13. Harvey, W., Stuckey, P.: A unit two variable per inequality integer constraint solver for constraint logic programming. In: The 20th Australasian Computer Science Conference (Australian Computer Science Communications), Sydney, Australia (1997) 102–111
14. Hsu, W., Nemhauser, G.: Algorithms for minimum covering by cliques and maximum clique in claw-free perfect graphs. Discrete Math. **37** (1981) 181–191
15. Hsu, W., Nemhauser, G.: A polynomial algorithm for the minimum weighted clique cover problem on claw-free perfect graphs. Discrete Math. **38**(1) (1982) 65–71
16. Jaffar, J., Maher, M., Stuckey, P., Yap, R.: Beyond finite domains. In: Proc. Principles and Practice of Constraint Programming. Volume 874 of Lect. Notes Comput. Sci. (1994) 86–94
17. Korte, B., Vygen, J.: Combinatorial Optimization: Theory and Algorithms. Springer–Verlag, Berlin (2011)
18. Lahiri, S., Musuvathi, M.: An efficient decision procedure for UTVPI constraints. In: Frontiers of Combining Systems. Volume 3717 of Lect. Notes Comput. Sci. (2005) 168–183
19. Maffray, F., Reed, B.: A description of claw-free perfect graphs. J. Combin. Theory, Ser. B **75**(1) (1999) 134–156
20. Padberg, M.: Perfect zero-one matrices. Math. Program. **6** (1974) 180–196
21. Peis, B.: Structure Analysis of Some Generalizations of Matchings and Matroids under Algorithmic Aspects. PhD thesis, Universität zu Köln (2007)
22. Schrijver, A.: Disjoint homotopic paths and trees in a planar graph. Discrete Comput. Geom. **6**(1) (1991) 527–574
23. Schrijver, A.: Combinatorial Optimization. Polyhedra and Efficiency (3 volumes). Volume 24 of Algorithms and Combinatorics. Springer–Verlag, Berlin (2003)
24. Schutt, A., Stuckey, P.: Incremental satisfiability and implication for UTVPI constraints. INFORMS J. Comput. **22**(4) (2010) 514–527
25. Seshia, S., Subramani, K., Bryant, R.: On solving Boolean combinations of UTVPI constraints. J. Satisf. Boolean Model. Comput. **3** (2007) 67–90
26. Tarjan, R.: Depth-first search and linear graph algorithms. SIAM J. Comput. **1**(2) (1972) 146–160
27. Whitesides, S.: A method for solving certain graph recognition and optimization problems, with applications to perfect graphs. In Berge, C., Chvátal, V., eds.: Topics on Perfect Graphs. Volume 88 of North-Holland Mathematics Studies. North-Holland (1984) 281–297