Fair Edge Deletion Problems on Tree-Decomposable Graphs and Improper Colorings[⋆]

Petr Kolman[1], Bernard Lidický[1], and Jean-Sébastien Sereni[1,2]

[1] Department of Applied Mathematics (KAM), Faculty of Mathematics and Physics
Charles University, Prague, Czech Republic
[2] CNRS (LIAFA, Université Denis Diderot) Paris, France

Abstract. In edge deletion problems, we are given a graph G and a graph property π and the task is to find a subset of edges the deletion of which results in a subgraph of G satisfying the property π. Typically the objective is to minimize the total number of deleted edges, while in less common fair versions the objective is to minimize the maximum number of edges removed from a single vertex. Since many fair edge deletion problems are NP-hard for general graphs, in the first part of the paper we restrict our attention to graphs with bounded tree-width, and in the second part we concentrate on approximation algorithms for two particular fair edge deletion problems on general graphs.

Many NP-hard problems become tractable when restricted to certain classes of graphs. The Monadic Second-Order Logic (MSOL) provides a rather general framework for dealing with NP-hard problems on graphs with bounded tree-width. A classical result by Courcelle shows that every problem expressible in MSOL can be solved in linear time for graphs with bounded tree-width; though this result was extended in different ways, none of them is suitable for fair edge deletion problems. Our main result is that every problem expressible in MSOL is solvable in polynomial time with the fair objective function, on graphs with bounded tree-width.

For general graphs, we focus our attention on the odd cycle transversal problem (the task is to make a given graph bipartite), again under the fair objective function; the problem being closely related to improper colorings of graphs. We describe a $\Theta(\sqrt{n})$-approximation algorithm. Analogous results hold for the minimum fair cut problem.

1 Introduction

Many problems in combinatorial optimization can be formulated as *edge* (or *node*) *deletion problems*. Given a graph $G = (V, E)$ and a graph property π (e.g., being a tree, a bipartite graph or a series-parallel graph), the problem is to find a subset of edges the deletion of which results in a subgraph of G satisfying the property π [31]. Typically, the objective function is to minimize the total number of deleted edges. (Since many edge deletion problems are NP-complete [5, 31], finding good approximation algorithms is an active and relevant area [2,

[⋆] This research was partially supported by the grant GA ČR 201/09/0197.

3].) In this paper we study edge deletion problems under a different objective function: our goal is to minimize the maximum number of edges removed from a single vertex (i.e., we want to minimize the maximum degree in (V, F) where F is the set of deleted edges); such problems are called *fair edge deletion problems*. For the fair objective function it is reasonable to deal also with, in some sense, complementary problems. Given a graph $G = (V, E)$ and a property π, find a subset F of edges such that (V, F) satisfies π and the degree of (V, F) is as small as possible; such problems are also called *fair edge deletion problems* in this paper. Informally, in both cases, instead of minimizing the total cost, we aim at minimizing the maximum cost at a vertex (in the first case, we have to pay for what is removed, in the second case for what remains). As we shall see, such problems have both a theoretical appeal and a relevance for practical applications.

Lin and Sahni [26] coined the term *fair edge deletion problems* when dealing with the following problem: given a graph $G = (V, E)$, find a subset F of edges such that $(V, E \setminus F)$ is a spanning tree, and the maximum degree of the graph (V, F) is as small as possible. They proved that the problem is NP-complete. Surprisingly enough, not many edge deletion problems were studied with the aforementioned *fair* objective function. Yet, such an objective function arises naturally in various contexts. As an example consider the (ℓ, k)-coloring problem: given a graph $G = (V, E)$ and integers ℓ and k, partition V into ℓ parts such that each part induces a subgraph of G with maximum degree at most k.

Graphs with bounded tree-width. Many NP-hard problems become tractable when restricted to certain classes of graphs. Since its introduction [7, 29], the class of graphs with bounded tree-width gained a considerable amount of attention, in particular with respect to the design of polynomial time algorithms for problems that are NP-hard on general graphs. The Monadic Second-Order Logic (see Sec. 2 for the definition) provides a rather general framework for dealing with NP-hard problems on graphs with bounded tree-width. It has been shown that every problem expressible in MSOL can be solved in linear time on graphs with bounded tree-width [4, 10, 15]. Although the MSOL is sufficient for dealing with many problems, many others do not fit in this framework. This motivated several extensions of the MSOL [11, 16].

In this paper, we go along those lines and extend the previous work in yet another direction that makes it possible to deal with several problems that do not fit into any of the previously known frameworks. Our main result shows that for properties expressible in MSOL, fair edge deletion problems are solvable in polynomial time on graphs of bounded tree-width (Section 3). To this end, we adopt and extend the techniques used by Borie et al. [12]. Though our proof does not use any substantially new techniques, the result is far from obvious and, as far as we know, was not known earlier.

As a corollary of our main result, we show, that for any fixed ℓ, the (ℓ, k)-coloring problem on graphs with bounded tree-width is solvable in polynomial time, where k is any integer. (Note that for any fixed k, the results of Rao [28] imply that the (ℓ, k)-coloring problem on graphs of bounded tree-width is solvable in polynomial time, where ℓ is any integer.)

General graphs. We focus on the fair odd cycle transversal problem. An *odd cycle transversal* (OCT) of a graph $G = (V, E)$ is a subset $F \subseteq E$ of edges such that the graph $(V, E \setminus F)$ is bipartite. The *minimum OCT* problem (also known as the odd-cycle edge cover [25], as the maximum cut problem [20] and as the minimum uncut problem [1]) consists in finding an OCT of minimum size. The *minimum fair OCT*, on the other hand, consists in finding an OCT F such that the maximum degree of (V, F) is as small as possible.

OCTs are closely related to *improper colorings* (also known as *defective colorings*). Note that a graph has a solution to the $(2, k)$-coloring problem if and only if the optimum value of the minimum fair OCT problem is at most k. Improper colorings are a natural generalization of the usual notion of a proper coloring, a core topic of the graph theory; they have been introduced and studied both as a theoretical notion [17, 23, 30] and as a tool to model practical problems [22]. Let us point out that even for planar graphs, the $(2, k)$-coloring problem is NP-complete [14, 18] and, thus, the minimum fair OCT is NP-hard on planar graphs. On the other hand, the minimum OCT problem is solvable in polynomial time on planar graphs [24] (though NP-hard for general graphs [19]). (Note that the minimum OCT is solvable in polynomial time also on graphs with bounded tree-width [9].)

The substantial difference between the complexity of the fair and the usual versions of the OCT for planar graphs provides a momentum to study the minimum fair OCT. Another reason to study the fair OCT is the lack of algorithms for the (ℓ, k)-coloring problems and, in particular, for the $(2, k)$-coloring problem.

We use linear programming to obtain a $\Theta(\sqrt{n})$-approximation algorithm for graphs on n vertices and we show that the bound on the performance of the algorithm is tight even for planar graphs (Section 4). Remarkably, the integrality gap of the linear program is $\Omega(n)$. As we also observe, all our results extend to the *minimum fair cut* problem: given a graph $G = (V, E)$ and two vertices $x, y \in V$, the problem consists in finding an x-y cut $F \subseteq E$ such that the maximum degree of (V, F) is as small as possible. Closely related is the *matching cut* problem: is there a cut F in G that is also a matching? Chvátal [13] proved that for general graphs, the problem is NP-hard.

Overview of the paper. Section 2 surveys the definitions and results about tree-decomposable graphs and the MSOL that we will need later. In Section 3 we deal with the fair edge deletion problems on tree-decomposable graphs and in Section 4 with approximation algorithms for the minimum fair OCT and s-t-cut problem on general graphs.

2 Tree-Decomposable Graphs and MSOL in a Nutshell

There are different yet more or less equivalent ways how to define tree-decomposable graphs [8]. Though probably the most common way today is that of Robertson and Seymour [29] (graphs of bounded tree-width), in this paper we decided to use the notion of k-terminal graphs as defined by Borie [10]; this choice will make it easier to link our results to previous work and present the proof of the new result in a short way. We stress again that a graph G has tree-width bounded by a constant if and only if G belongs to a class of tree-decomposable graphs as defined below.

A *k-terminal graph* $G = (V, E, T)$ is a graph $G = (V, E)$ with an ordered set T of at most k distinguished vertices from V called *terminals* (the others being the *non-terminals*). We stress that the number of terminals in a k-terminal graph may be *smaller* than k.

A *k-terminal graph composition operation* constructs a new k-terminal graph from several k-terminal graphs. A k-terminal composition operation $\oplus$ of arity m takes as arguments m k-terminal graphs $G_1 = (V_1, E_1, T_1), \ldots, G_m = (V_m, E_m, T_m)$ and out of them produces a new graph G with k' terminals, for some $k' \leq k$. The new graph is obtained by merging some of the terminal nodes of the graphs $G_1, \ldots, G_m$ together. In particular, the operation $\oplus$ specifies a constant $l_\oplus \geq 0$ and

- for each $i \in \{1, \ldots, k'\}$, a set $X_i \subseteq \cup_{j=1}^{m} T_j$ such that $|X_i \cap T_j| \leqslant 1$ for each $j \in \{1, ..., m\}$; the i^{th} terminal of G is then produced by identifying the terminals in X_i;
- for each $i \in \{1, \ldots, l_\oplus\}$, a set $X'_i \subseteq \cup_{j=1}^{m} T_j$ such that $|X'_i \cap T_j| \leqslant 1$ for each $j \in \{1, \ldots, m\}$; the i^{th} new non-terminal of G is then produced by identifying the terminals in X'_i.

Moreover, the sets X_i for $i \in \{1, \ldots, m\}$ and X'_i for $i \in \{1, \ldots, l_\oplus\}$ must be pairwise disjoint.

We represent the operation $\oplus$ by a matrix $M_\oplus$ of size $(k' + l_\oplus) \times m$ in the following way: if $i \leq k'$ then $M_\oplus[i, j]$ is the index of the terminal of G_j that belongs to X_i if there is one, and 0 otherwise; if $i \in \{k' + 1, \ldots, k' + l_\oplus\}$ then $M_\oplus[i, j]$ is the index of the terminal of G_j that belongs to X'_i if there is one, and 0 otherwise. For a vertex $v \in V_1 \cup \ldots \cup V_m$ and for an edge $e \in E_1 \cup \ldots \cup E_m$, it is convenient to define $\sigma_\oplus(v)$ and $\sigma_\oplus(e)$ to be the vertex and the edge of G to which v and e are mapped when we apply the operation $\oplus(G_1, \ldots, G_m)$, respectively. For a subset S of vertices or edges, $\sigma(S) = \bigcup_{x \in S} \{\sigma(x)\}$.

Let B be a finite set of k-terminal graphs and let R be a finite set of composition operations for k-terminal graphs; we call the graphs in B *base graphs*. We define the *k-terminal recursive family* (B, R) to be the closure of B by operations in R. A class of graphs C is said to be a class of *tree-decomposable graphs* if there exist two sets B and R such that $C = (B, R)$. A graph $G = (V, E)$ from a class $C = (B, R)$ can be represented by a rooted labeled tree (a *decomposition tree* of G) of size $O(|V|)$ with leaves labeled by graphs from the set of base graphs B and inner nodes labeled by composition operations from the set R. In this way every node v of the decomposition tree of G corresponds to a unique graph G_v from the class C.

The *Monadic Second-Order Logic (MSOL)* for graphs contains variables of four different *types* (sorts): $\mathbf{v}, \mathbf{e}, \mathbf{V}$ and $\mathbf{E}$, that is, variables for vertices, edges, subsets of vertices and subsets of edges, respectively. If we need to specify the type of a variable by its name, we adopt the convention that v_i, e_i, V_i and E_i denote variables of types $\mathbf{v}, \mathbf{e}, \mathbf{V}$ and $\mathbf{E}$, respectively. The set of primitive MSOL predicates[1] consists of $v_i = v_j$, $\text{Incident}(v_i, e_j)$, $v_i \in V_j$ and $e_i \in E_j$ with the obvious meanings. In *counting MSOL* [16], there are the additional primitive predicates $|V_i| = a \pmod b$ and $|E_i| = a \pmod b$, with $b \geq 2$ and $0 \leq a < b$. If P and Q are (counting) MSOL predicates, then so are $(\neg P)$, $(P \vee Q)$ and $(P \wedge Q)$. If P is a (counting) MSOL predicate with a free variable x, then so are $(\exists x)(P(x))$ and $(\forall x)(P(x))$.

Let $P(x_1, \ldots, x_t)$ be a (counting) MSOL predicate. For a graph $G = (V, E)$, we let D_i be the domain of the variable x_i in the model G, that is, D_i is the set V, E, 2^V or 2^E, depending on the type $\mathbf{v}$, $\mathbf{e}$, $\mathbf{V}$ or $\mathbf{E}$ of x_i, respectively. Then, given $y_1 \in D_1, \ldots, y_t \in D_t$, we define $P_G(y_1, \ldots, y_t)$ to be the truth value of the predicate $P(x_1, \ldots, x_t)$ in the model G. To distinguish between a (name of a) variable and a value of a variable, we use x_i for the former and y_i for the later (most of the time). Let x_i be a variable with domain D. Let $\star_{\mathbf{v}}$ and $\star_{\mathbf{e}}$ be two symbols that do not appear in V and E, respectively. We define $D^\star$ to be $D \cup \{\star_{\mathbf{v}}\}$ if the type of x_i is $\mathbf{v}$, $D \cup \{\star_{\mathbf{e}}\}$ if the type of x_i is $\mathbf{e}$, and D otherwise.

Let $C = (B, R)$ be a class of tree-decomposable graphs and let $P(x_1, \ldots, x_t)$ be a (counting) MSOL predicate with free variables $x_1, \ldots, x_t$. A key concept in the work of Borie et al. [12] (cf. [6]) is that of an equivalence classe of the set

$$\mathcal{B}_P^C = \{((V, E), y_1, \ldots, y_t) \,|\, (V, E) \in C, y_1 \in D_1^\star, \ldots, y_t \in D_t^\star\}$$

where D_i is the domain of x_i with respect to (V, E).

We say that $(G_1, y_{11}, \ldots, y_{1t}), \ldots, (G_m, y_{m1} \ldots, y_{mt}) \in \mathcal{B}_P^C$ are *compatible with respect to the (m-ary) operation $\oplus \in R$ and the variable x_ℓ of the predicate* $P(x_1, \ldots, x_t)$, if the following holds (letting V_i and E_i be the vertex set and the edge set of G_i, respectively).

– If x_ℓ is of type $\mathbf{v}$, then either
(1.1) there exists an integer i such that for each $j \in \{1, \ldots, m\}$, if $M_\oplus[i, j] \neq 0$ then $y_{j\ell}$ is the $M_\oplus[i, j]$-th terminal of G_j, or
(1.2) $y_{1\ell} = y_{2\ell} = \ldots = y_{m\ell} = \star_{\mathbf{v}}$, or
(1.3) there exists j such that $y_{j\ell} \in V_j$ and $y_{j'\ell} = \star_{\mathbf{v}}$ for $j' \neq j$;
– if x_ℓ is of type $\mathbf{e}$, then either
(2.1) $y_{1\ell} = y_{2\ell} = \ldots = y_{m\ell} = \star_{\mathbf{e}}$, or
(2.2) there exists j such that $y_{j\ell} \in E_j$ and $y_{j'\ell} = \star_{\mathbf{e}}$ for $j' \neq j$;
– if x_ℓ is of type $\mathbf{V}$, then
(3.1) for each $j \in \{1, \ldots, m\}$, if $y_{j\ell}$ contains a terminal of G_j that participates in the birth of the i-th terminal (or i-th non-terminal) of $\oplus(G_1, \ldots, G_m)$, then for each $j' \neq j$, $y_{j'\ell}$ contains the $M_\oplus[i, j']$-th terminal (or $M_\oplus[k' + i, j']$-th terminal, respectively) of $G_{j'}$ where k' is the number of terminals of $\oplus(G_1, \ldots, G_m)$;

[1] Borie *et al.* use the name *regular* predicates instead of MSOL predicates.

We note that for each type of variable, the considered cases are mutually disjunctive. The elements $(G_1, y_{11}, \dots, y_{1t}), \dots, (G_m, y_{m1} \dots, y_{mt}) \in \mathcal{B}_P^C$ are *compatible with respect to an m-ary operation $\oplus \in R$ and the predicate $P(x_1, \dots, x_t)$* if they are compatible with respect to every variable of $P(x_1, \dots, x_t)$.

For every (m-ary) operation $\oplus \in R$ we now define an extension $\bar{\oplus}$ that assigns to every compatible m-tuple $((G_1, y_{11}, \dots, y_{1t}), \dots, (G_m, y_{m1} \dots, y_{mt}))$ an element from $\mathcal{B}_P^C$. In particular, $\bar{\oplus}((G_1, y_{11}, \dots, y_{1t}), \dots, (G_m, y_{m1} \dots, y_{mt})) = (G, y_1, \dots, y_t)$ where $G = \oplus(G_1, \dots, G_m)$ and y_l, for $1 \leq \ell \leq t$, is defined as follows:

- if x_ℓ is of type $\mathbf{v}$, then in the case (1.1), y_ℓ is the i-th terminal of G, in the case (1.2), $y_\ell = \star_{\mathbf{v}}$, and in the case (1.3), $y_\ell = y_{jl}$;
- if x_ℓ is of type $\mathbf{e}$, then in the case (2.1), $y_\ell = \star_{\mathbf{e}}$ and $y_\ell = y_{j\ell}$ in the case (2.2);
- if x_ℓ is of type $\mathbf{V}$, then $y_\ell = \bigcup_{j=1}^{m} \sigma(y_{j\ell})$;
- if x_ℓ is of type $\mathbf{E}$, then $y_\ell = \bigcup_{j=1}^{m} \sigma(y_{j\ell})$.

The following theorem is implicit in the work of Borie et al. [12].

Theorem 1 (Borie et al. [12]). *If $P = (x_1, \dots, x_t)$ is a (counting) MSOL predicate and $C = (B, R)$ is a class of tree-decomposable graphs, then there exist:*

- *a finite set Q, and,*
- *for each composition operation $\oplus \in R$ an operation $\bigoplus$ on Q, of the same arity as $\oplus$, and,*
- *a function $h : \mathcal{B}_P^C \to Q$ such that*
  - *$h(G, y_1, \dots, y_t) = h(G', y'_1, \dots, y'_t)$ implies $P_G(y_1, \dots, y_t) = P_{G'}(y'_1, \dots, y'_t)$, whenever both expressions $P_G(y_1, \dots, y_t)$ and $P_{G'}(y'_1, \dots, y'_t)$ are defined, and,*
  - *for every $\oplus \in R$ (of arity m) and every compatible $((G_i, y_{i1}, \dots, y_{it}))_{1 \leqslant i \leqslant m}$*

$$\begin{aligned} &h(\overline{\oplus}((G_1, y_{11}, \dots, y_{1t}), \dots, (G_m, y_{m1}, \dots, y_{mt}))) \\ &\qquad = \bigoplus(h(G_1, y_{11}, \dots, y_{1t}), \dots, h(G_m, y_{m1}, \dots, y_{mt})) \,. \end{aligned} \tag{1}$$

The function h is called a *homomorphism* and the set Q a set of *equivalence classes* [2]. A class $q \in Q$ is *accepting* if $h(G, y_1, \dots, y_t) = c$ implies that $P_G(y_1, \dots, y_t)$ is true.

The proof of the theorem also provides an efficient way of constructing the operations $\bigoplus$. Thus, a corollary of the theorem is an alternative proof of the famous result, mentioned in the introduction, that every graph problem expressible in (counting) MSOL (also called *recognition* problem) can be solved in polynomial (even linear) time for tree-decomposable graphs: given a graph $G \in C$, construct a decomposition tree J of G and use (1) to evaluate the function h in a bottom-up manner for all nodes of the decomposition tree J and all relevant y-values; with some effort one needs at most $O(1)$ time per node

[2] For more details we refer to the papers [6, 12].

of the decomposition tree, for $k = O(1)$. Borie et al. describe how to employ Theorem 1 to obtain linear time algorithms also for *optimization* problems (i.e., $\min \{|y| \mid P_G(y)\}$ or $\max \{|y| \mid P_G(y)\}$) and *enumeration* problems (i.e., computing $|\{(y_1, \ldots, y_t) \mid P_G(y_1, \ldots, y_t)\}|$).

Theorem 2 (Arnborg et al. [4], Borie et al. [12], Courcelle [15], Courcelle and Mosbah [16]). *The recognition, optimization and enumeration problems can be solved in linear time on tree-decomposable graphs.*

3 Fair Edge Deletion Problems on Graphs with Bounded Tree-width

We describe how to exploit the MSOL framework to solve in polynomial time fair edge deletion problems on tree-decomposable graphs. In the proof of the main theorem of this section, we exploit and extend the homomorphism-based method [12].

For a graph $G = (V, E)$ and a subset $y \subseteq E$, we let Δ_y be the maximum degree in the graph (V, y).

Theorem 3. *If $P(x)$ is a (counting) MSOL predicate with a free variable of type* $\mathbf{E}$*, then the problem*

$$\min_{y \subseteq E} \left\{ \Delta_y \mid P_{(V,E)}(y) \right\}$$

is solvable in polynomial time on tree-decomposable graphs.

Proof. Let C be a fixed class of k-terminal tree-decomposable graphs. Let Q be the finite set of equivalence classes and $h : \mathcal{B}_P^C \to Q$ the homomorphism given by Theorem 1. Let $G = (V, E) \in C$, let Δ be the maximum degree of G and H a decomposition tree of G. For a node v of H, let $G_v = (V_v, E_v)$ be the graph corresponding to this node (i.e., if v is a leaf and H is the label of v, then $G_v = H$; if v is an inner node, $\oplus$ is the label of v and $H_1, \ldots, H_m$ are the graphs corresponding to the children of v, then $G_v = \oplus(H_1, \ldots, H_m)$) and let $T_v = \{t_1^v, \ldots, t_{k'}^v\}$ be the set of at most k terminals of G_v and $A_v = E_v \cap \binom{T_v}{2}$ the set of edges between terminals of G_v. Our intention is to compute for every node v of the decomposition tree H and every equivalence class $q \in Q$ a matrix M_q^v of dimension $k' + 1$ and size $2^{|A_v|} \cdot (\Delta + 1)^{k'}$ where the first dimension is indexed by subsets of A_v and all other dimensions by integers between 0 and Δ. For $A \subseteq A_v$ and k' integers $i_1, \ldots, i_{k'}$ between 0 and Δ, we define

$$\begin{aligned} &M_q^v(A, i_1, \ldots, i_{k'}) := \\ &\quad \min \left\{ \Delta_y \,\middle|\, y \subseteq E_v, y \cap A_v = A, h(G_v, y) = q, \deg_y(t_j^v) \le i_j, 1 \le j \le k' \right\} \quad (2) \end{aligned}$$

where $\deg_y(u)$ is the degree of u in the graph (V_v, y). With such matrices at hand, a solution for the problem readily follows: the optimal solution is

$$\min_{q \in Q_a} \min_{A \subseteq A_{root}} M_q^{root}(A, \Delta, \ldots, \Delta)$$

where $Q_a \subseteq Q$ is a set of accepting classes. It remains to describe how to obtain such matrices.

For leaves of the decomposition trees (i.e., for the base graphs in B) the matrices can be computed by brute force. Let v be a fixed inner node of the decomposition tree and $\oplus$ its label. Since the composition operation $\oplus$ provides an exact description of the construction of G_v out of the graphs associated with the children of v in H, the matrices associated with the children of v contain sufficient information for calculation of the matrix M_q^v, for every equivalence class $q \in Q$. It is easy to see that the time to compute a single value of the matrix is bounded by a polynomial in $|V|$ and, thus, also the time to compute the complete matrix M_q^v is bounded by a polynomial. Since the number of equivalence classes is constant and the size of the decomposition tree is linear in $|V|$, we conclude that the total time to compute all the matrices for all nodes of the decomposition tree is bounded by polynomial in $|V|$. □

We note that, since the time for calculation of M_q^v grows with the degree of the decomposition tree, it is desirable to have the degrees as small as possible.

We conclude this section by observing that a straightforward modification of the procedure permits to obtain not only the value of the optimal solution but also a corresponding set.

3.1 Examples

In this subsection we deal with two examples of fair edge deletion problems: minimum degree spanning tree and (ℓ, k)-coloring. An exact ad-hoc polynomial algorithm for the minimum degree spanning tree was given by Marathe et al. [27]; for known results about the (ℓ, k)-coloring we refer to the Introduction.

The predicates "connected" and "forest" are known to be MSOL predicates (cf. [12]). Therefore also the predicate "spanning tree" is a MSOL predicate and thus, by Theorem 3, the minimum degree spanning tree problem is solvable in polynomial time on tree-decomposable graphs.

An (ℓ, k)-*coloring* of a graph $G = (V, E)$ is a partition of V into ℓ parts such that each part induces a subgraph of G with maximum degree at most k. The problem that we consider here is the following: For a graph $G = (V, E)$ and a fixed value ℓ the task is to find the minimum k such that G has an (ℓ, k)-coloring. Consider the following predicate:

$$P(F') = (\exists V_1, \ldots, V_\ell, F) F = E \backslash F' \wedge \text{partition}(V_1, \ldots, V_\ell, V) \wedge \bigwedge_{i=1}^{\ell} (V_i \times V_i \cap F = \emptyset)\,,$$

where the predicate "partition" is true if the sets V_i specify a partition of V; "partition" is known to belong to the MSOL. Then, P is an MSOL predicate, and hence Theorem 3 ensures that the (ℓ, k)-coloring problem is solvable in polynomial time on tree-decomposable graphs, which is a new result.

4 Approximation of Minimum Fair OCT and Minimum Fair *s*-*t* Cut on General Graphs

Given a graph $G = (V, E)$, let $\mathcal{C}$ be the set of all odd cycles (viewed as edge sets). The minimum fair OCT problem can be formulated using integer linear programming as follows.

$$\begin{aligned} & \text{minimize} \quad k \qquad\qquad (3) \\ \text{subject to} \quad & \forall C \in \mathcal{C}, \quad \sum_{e \in C} x_e \geq 1\,, \\ & \forall u \in V, \quad \sum_{e \in \delta(v)} x_e \leq k\,, \\ & \forall e \in E, \quad x_e \in \{0,1\}\,. \end{aligned}$$

In a linear programming relaxation, the last condition is replaced by $x_e \geq 0$.

We pause to note that the number of inequalities in the linear program (LP) may be exponential in the size of G, nevertheless, the LP is solvable in polynomial time using the ellipsoid method (given a vector x, one can check whether there is a violated inequality [21]).

We also notice that the integrality gap of the relaxation by itself is very large, namely n, as it can be seen by considering an odd cycle of length n: the fractional optimum is $2/n$ (for each edge we set $x_e = 1/n$) while the integral optimum is 1. We now combine the relaxation with a few observations to obtain an $O(\sqrt{n})$-approximation.

Solve the LP for the graph $G = (V, E)$, and set $F = \{e \in E \,|\, x_e \geq 1/(4\sqrt{n})\}$ and $\overline{F} = E \setminus F$. Let H be the subset of edges of E that have both end degrees at most $\sqrt{n}+1$ in $(V, \overline{F})$, that is, $H = \big\{\{u,v\} \in \overline{F} \,\big|\, \deg_{\overline{F}}(u) \leq \sqrt{n}+1,\ \deg_{\overline{F}}(v) \leq \sqrt{n}+1\big\}$.

Lemma 1. *The graph $G' = (V, \overline{F} \setminus H))$ is bipartite.*

Proof. Suppose on the contrary that G' contains an odd cycle, and let C be a shortest one. First, note that the length of C is at least $4\sqrt{n}+1$: since x is a feasible solution of the LP for G', every odd cycle of length at most $4\sqrt{n}$ in G' contains an edge e with $x_e \geq 1/(4\sqrt{n})$.

Let D be the set of edges with exactly one endpoint in C; note that the only edges with two endpoints in C are edges of the cycle. Since at least one of every two successive nodes in C has at least $\sqrt{n}+2$ neighbors and since at least $\sqrt{n}$ of these neighbors are outside C, the set D contains $\sqrt{n} \cdot 4\sqrt{n}/2 = 2n$ or more edges. On the other hand, since C is the shortest odd cycle, every vertex outside C has at most two neighbors in C. Hence, $|D| \leqslant 2(n - 4\sqrt{n}+1) < 2n$; a contradiction. □

Theorem 4. *The above procedure computes an $O(\sqrt{n})$-approximation of the minimum fair odd cycle transversal.*

Proof. Let $\hat{k}$ be the integral optimum and k^* the fractional optimum (so $k^* \leq \hat{k}$). We assume that the input graph G is not bipartite, that is, $\hat{k} \geq 1$. Then, for every vertex $u \in V$ the set H contains at most $\sqrt{n}+1$ edges adjacent to u. Moreover, for every vertex $u \in V$ the rounding procedure guarantees that the number of edges from F adjacent to u is at most $4\sqrt{n}\sum_{e\in\delta(u)} x_e \leq 4\sqrt{n}k^*$. □

A simple example demonstrates that the given bound on the approximation ratio of our algorithm is tight even for planar (and series-parallel) graphs. Think about a graph obtained from a cycle of length $2\sqrt{n}+1$ by replacing all edges but one by $\sqrt{n}/2$ internally vertex disjoint paths of length two. The minimum value of an OCT is one while the value of an OCT reported by the algorithm is $\sqrt{n}$.

We conclude this section with an observation that the integrality gap of the LP is large even for planar (and series-parallel) graphs.

Theorem 5. *The integrality gap of* (3) *for* 2*-connected planar (and series-parallel) graphs is* $\Omega(\sqrt{n})$.

Proof. Let n be an integer such that $\sqrt{n}$ is an integer. Consider $\sqrt{n}$ vertices $v_1, v_2, \ldots, v_{\sqrt{n}}$. For each $i \in \{1, 2, \ldots, \sqrt{n}-1\}$, add $\sqrt{n}$ parallel edges between v_i and v_{i+1}, and subdivide each of these edges once. Thus, v_i and v_{i+1} are linked by $\sqrt{n}$ internally disjoint paths $P^i_1, \ldots, P^i_{\sqrt{n}}$ of length 2. Last, add an edge between $v_{\sqrt{n}}$ and v_i for every $i \in \{1, \ldots, \lceil\sqrt{n}/2\rceil\}$. Let $G = (V, E)$ be the obtained graph (see Figure 4). Thus, G is a planar graph with n vertices and maximum degree $2\sqrt{n}+1$. Further, every odd cycle of G contains the vertex $v_{\sqrt{n}}$ and has length at least $2 \cdot \lfloor\sqrt{n}/2\rfloor + 1$. Thus, setting $x_e = 1/\sqrt{n}$ for every edge e of G yields a feasible fractional solution with objective value $2 + \frac{1}{\sqrt{n}}$. However, the integral optimum is $\lceil\sqrt{n}/2\rceil$. To see this, let F be an OCT of G. If all the edges $\{v_{\sqrt{n}}, v_i\}$ for $i \in \{1, \ldots, \lceil\sqrt{n}/2\rceil\}$ belong to F, then the degree of $v_{\sqrt{n}}$ in (V, F) is at least $\lceil\sqrt{n}/2\rceil$, as wanted. So, assume that there exists $i \in \{1, \ldots, \lceil\sqrt{n/2}\rceil\}$ such that $\{v_{\sqrt{n}}, v_i\} \notin F$. Then, observe that there must exist $j \in \{i, \ldots, \sqrt{n}\}$ such that F contains at least one edge of each of the paths P^j_s. (For otherwise there would exist an odd cycle $v_i, y_i, v_{i+1}, y_{i+1}, \ldots, v_{\sqrt{n}-1}, y_{\sqrt{n}-1}, v_{\sqrt{n}}$ in $(V, E \setminus F)$, where y_t belongs to P^t_s for some integer s.) Consequently, one of v_j and v_{j+1} has degree at least $\lceil\sqrt{n}/2\rceil$ in the graph (V, F). This concludes the proof. □

Minimum Fair Cut Problem. The same approach yields a $\Theta(\sqrt{n})$-approximation algorithm for the minimum fair cut problem; the difference is that in the linear program (3) we replace the set $\mathcal{C}$ of all odd cycles by the set of all paths between x and y. We omit further details.

5 Conclusion and Open Problems

The approach presented in this paper can be extended to other *fair* objective functions; an example are fair node deletion problems where the goal is to delete a proper subset of nodes such that the resulting graph satisfies the given property and the objective is roughly to minimize the maximum number of

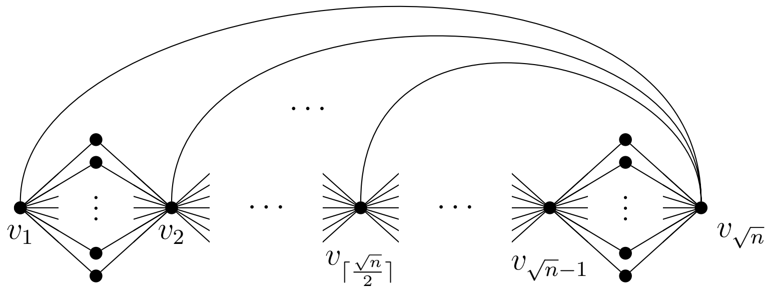

Fig. 1. The graph G in the proof of Theorem 5.

deleted neighbors. As we alluded to, the approach also works for maximizing the minimum degree. The question is for which other objective functions this or a similar approach works?

An important open question deals with the complexity of the fair problems. The running time of the exact algorithms for tree-decomposable graphs is polynomial. Is it possible to have a linear-time algorithm?

References

1. A. Agarwal, M. Charikar, K. Makarychev, and Y. Makarychev. $O(\sqrt{\log n})$ approximation algorithms for min UnCut, min 2CNF deletion, and directed cut problems. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC)*, pages 573–581, 2005.
2. N. Alon, A. Shapira, and B. Sudakov. Additive approximation for edge-deletion problems. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 419–428, 2005.
3. N. Alon, A. Shapira, and B. Sudakov. Additive approximation for edge-deletion problems. *Ann. Math.*, 170:371–411, 2009.
4. S. Arnborg, J. Lagergren, and D. Seese. Problems easy for tree-decomposable graphs. *Journal of Algorithms*, 12:308–340, 1991.
5. T. Asano. An application of duality to edge-deletion problems. *SIAM Journal on Computing*, 16(2):312–331, 1987.
6. M. W. Bern, E. L. Lawler, and A. L. Wong. Linear-time computation of optimal subgraphs of decomposable graphs. *Journal of Algorithms*, 8(2):216–235, 1987.
7. H. L. Bodlaender. A tourist guide through treewidth. *Acta Cybernet.*, 11(1-2):1–21, 1993.
8. H. L. Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theoret. Comput. Sci.*, 209(1-2):1–45, 1998.
9. H. L. Bodlaender and K. Jansen. On the complexity of the maximum cut problem. In *Nordic Journal of Computing*, volume 7. 2000.
10. R. B. Borie. *Recursively Constructed Graph Families: Membership and Linear Algorithms.* PhD thesis, Georgia Inst. Tech., Dept. Computer Science, 1988.
11. R. B. Borie. Generation of polynomial-time algorithms for some optimization problems on tree-decomposable graphs. *Algorithmica*, 14(2):123–137, 1995.

12. R. B. Borie, R. G. Parker, and C. A. Tovey. Automatic generation of linear-time algorithms from predicate calculus descriptions of problems on recursively constructed graph families. *Algorithmica*, 7:555–581, 1992.
13. V. Chvátal. Recognizing decomposable graphs. *J. Graph Theory*, 8(1):51–53, 1984.
14. R. Corrêa, F. Havet, and J.-S. Sereni. About a Brooks-type theorem for improper colouring. *Australas. J. Combin.*, 43:219–230, 2009.
15. B. Courcelle. The monadic second-order theory of graphs. I. recognizable sets of finite graphs. *Information and Computation*, 85(1):12–75, 1990.
16. B. Courcelle and M. Mosbah. Monadic second-order evaluations on tree-decomposable graphs. *Theoretical Computer Science*, 109(1–2):49–82, 1 Mar. 1993.
17. L. J. Cowen, R. H. Cowen, and D. R. Woodall. Defective colorings of graphs in surfaces: partitions into subgraphs of bounded valency. *J. Graph Theory*, 10(2):187–195, 1986.
18. L. J. Cowen, W. Goddard, and C. E. Jesurum. Defective coloring revisited. *J. Graph Theory*, 24(3):205–219, 1997.
19. M. Garey, D. S. Johnson, and L. J. Stockmeyer. Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1:237–267, 1976.
20. M. X. Goemans and D. P. Williamson. .879-approximation algorithms for MAX CUT and MAX 2SAT. In *Proceedings of the 26th Annual ACM Symposium on the Theory of Computing (STOC)*, pages 422–431, 1994.
21. M. Groetschel and W. R. Pulleyblank. Weakly bipartite graphs and the max-cut problem. *Operations Research Letters*, 1(1):23 – 27, 1981.
22. F. Havet, R. J. Kang, and J.-S. Sereni. Improper colouring of unit disk graphs. *Networks*, 54(3):150–164, 2009.
23. F. Havet and J.-S. Sereni. Improper choosability of graphs and maximum average degree. *J. Graph Theory*, 52(3):181–199, 2006.
24. D. S. Johnson. The NP-completeness column: An ongoing guide. *Journal of Algorithms*, 6(1):145–159, 1985.
25. J. Komlós. Covering odd cycles. *Combinatorica*, 17(3):393–400, 1997.
26. L.-S. Lin and S. Sahni. Fair edge deletion problems. *IEEE Trans. Computers*, 38(5):756–761, 1989.
27. M. V. Marathe, R. Ravi, R. Sundaram, S. S. Ravi, D. J. Rosenkrantz, and H. B. Hunt III. Bicriteria network design problems. *Journal of Algorithms*, 28(1):142–171, 1998.
28. M. Rao. MSOL partitioning problems on graphs of bounded treewidth and clique-width. *Theoret. Comput. Sci.*, 377(1-3):260–267, 2007.
29. N. Robertson and P. D. Seymour. Graph Minors II. algorithmic aspects of tree width. *Journal of Algorithms*, 7:309–322, 1986.
30. D. R. Woodall. Defective choosability of graphs with no edge-plus-independent-set minor. *J. Graph Theory*, 45(1):51–56, 2004.
31. M. Yannakakis. Edge-deletion problems. *SIAM Journal on Computing*, 10(2):297–309, 1981.