Complexity and Approximability for Parameterized CSPs

Holger Dell[1], Eunjung Kim[2], Michael Lampis[3], Valia Mitsou[*4], and Tobias Mömke[†5]

1 **Saarland University and Cluster of Excellence (MMCI), Saarbrücken**
`hdell@mmci.uni-saarland.de`
2,3 **Université Paris Dauphine**
`eunjungkim78@gmail.com, michail.lampis@dauphine.fr`
4 **SZTAKI, Hungarian Academy of Sciences, Budapest**
`vmitsou@gradcenter.cuny.edu`
5 **Saarland University, Saarbrücken**
`moemke@cs.uni-saarland.de`

Abstract

The complexity of various Constraint Satisfaction Problems (CSP) when parameterized by structural measures (such as treewidth or clique-width) is a very well-investigated area. In this paper, we take a fresh look at such questions from the point of view of approximation, considering four standard CSP predicates: AND, OR, PARITY, and MAJORITY. By providing new or tighter hardness results for the satisfiability versions, as well as approximation algorithms for the corresponding maximization problems, we show that already these basic predicates display a diverse set of behaviors, ranging from being FPT to optimize exactly for quite general parameters (PARITY), to being W-hard but well-approximable (OR, MAJORITY) to being W-hard and inapproximable (AND). Our results indicate the interest in posing the question of approximability during the usual investigation of CSP complexity with regards to the landscape of structural parameters.

1998 ACM Subject Classification F.2.0

Keywords and phrases Approximation, Structural Parameters, Constraint Satisfaction

Digital Object Identifier 10.4230/LIPIcs.xxx.yyy.p

1 Introduction

Constraint Satisfaction Problems (CSPs) play a central role in almost all branches of theoretical computer science. Starting from CNFSAT, the prototypical NP-complete problem, the computational complexity of CSPs has been widely studied from various points of view. In this paper we focus on two aspects of CSP complexity which, though extremely well-investigated, have mostly been considered separately so far in the literature: parameterized complexity and approximability. We study four standard predicates and contribute some of the first results indicating that the point of view of approximability considerably enriches the parameterized complexity landscape of CSPs.

Parameterized CSPs. The vast majority of interesting CSPs are NP-hard [28, 18]. This has motivated the study of such problems from a parameterized complexity point of view, and

* Supported by ERC Starting Grant PARAMTIGHT (No. 280152)
† This research is supported by Deutsche Forschungsgemeinschaft grant BL511/10-1

indeed this topic has attracted considerable attention in the literature [14, 32, 11, 25, 13, 31]. We refer the reader to [27] where an extensive classification of CSP problems for a large range of parameters is given. In this paper we focus on *structurally* parameterized CSPs, that is, we consider CSPs where the parameter is some measure of the structure of the input instance. The central idea behind this approach is to represent the structure of the CSP using a (hyper-)graph and leverage the powerful tools commonly applied to parameterized graph problems (such as tree decompositions) to solve the CSP.

The typical goal of this line of research is to find the most general parameterization of a CSP that still remains fixed-parameter tractable (FPT). To give a concrete example for a very well-known CSP, CNFSAT is FPT when parameterized by the treewidth of its incidence graph[1] but it is W-hard for more general parameters such as clique-width [23], or even the more restricted modular treewidth [24]. General (boolean) CSP on the other hand, where the description of each constraint is part of the input is known to be a harder problem: it is already W[1]-hard parameterized by the incidence treewidth, but FPT parameterized by the treewidth of the primal graph [30]. Thus, parameterized investigations aim to locate the boundary where a CSP jumps from being FPT to being W-hard. It is of course a natural question how we can deal with the W-hard cases of a CSP once they are identified.

Approximation. CSPs also play a central role in the theory of (polynomial-time) approximation algorithms [33, 19, 4]. In this context we typically consider a CSP as an optimization problem (MAXCSP) where the goal is to find an assignment to the variables that satisfies as many of the constraints as possible. Unfortunately, essentially all non-trivial CSPs are hard to approximate (APX-hard) from this point of view [18], even those where deciding if an assignment can satisfy all constraints is in P (e.g. 2CNFSAT or Horn SAT). Thus, research in this area typically focuses on discovering exactly the best approximation ratio that can be achieved in polynomial time. Amazingly, for many natural CSPs this happens to be exactly the ratio achieved by a completely random assignment [17]. This motivates the question of whether we can find natural cases where non-trivial efficient approximations are possible.

Results. In this paper we consider four different types of CSPs where the constraints are respectively OR, AND, PARITY and MAJORITY functions. Our approach follows, for the most part, the standard parameterized complexity script: we consider the input instance's incidence graph and try to determine the complexity of the CSP when parameterized by various graph widths. The new ingredient in our approach is that, in addition to trying to determine which parameters make a CSP FPT or W-hard, we also ask if the optimization versions of W-hard cases can be well-approximated. We believe that this is a question of special interest since, as it turns out, there are CSPs for which W-hardness can be (almost) circumvented using approximation, and others which are inapproximable.

More specifically, our results are as follows: for OR constraints, which corresponds to the standard CNFSAT (MAXCNFSAT) problem, we present a new hardness proof establishing that deciding a formula's satisfiability is W-hard even if parameterized by the incidence graph's neighborhood diversity. Neighborhood diversity is a parameter much more restricted than modular treewidth [20], for which the strongest previously known W[1]-hardness result was known [24]. We complement this negative result with a strong positive approximation result: there exists an FPT Approximation Scheme (FPT-AS)[2] for MAXCNFSAT parameterized by clique-width, that is, an algorithm which for all $\epsilon > 0$ runs in time $f(k,\epsilon)n^{O(1)}$ and returns an assignment satisfying $(1-\epsilon)$OPT clauses. Thus, even though we establish that

[1] See the next section for a definition of incidence graphs
[2] We follow here the standard definition of FPT-AS given in [22].

solving CNFSAT exactly is W-hard even for extremely restricted dense graph parameters, MAXCNFSAT is well-approximable even in the quite general case of clique-width. To the best of our knowledge, this is the first approximation result of this type for a W-hard MAXCSP problem.

Recalling that CNFSAT is FPT parameterized by the treewidth of the incidence graph, we consider other problems for which the jump from treewidth to clique-width could have interesting complexity consequences. We show that AND and PARITY constraints exhibit two wildly different behaviors. On the one hand, the problem of maximizing the largest possible number of satisfied PARITY constraints remains FPT even for dense parameters such as clique-width. On the other hand, by modifying our reduction for CNFSAT, we are able to show not only that maximizing the number of satisfied AND constraints is W[1]-hard parameterized by neighborhood diversity, but also that this problem cannot even admit an FPT-AS (like MAXCNFSAT), unless W[1]=FPT. We recall that PARITY and AND constraints are similar in other aspects: for example, for both we can decide in polynomial time if an assignment satisfying all constraints exists.

Finally, we consider CSPs with MAJORITY constraints, that is, constraints which are satisfied if at least half their literals are true. We give a reduction establishing that this is an interesting case of a natural constraint type for which deciding satisfiability is already W[1]-hard parameterized by incidence treewidth. We complement this negative result with two algorithmic results: first, we show that the corresponding MAXCSP is FPT parameterized by incidence vertex cover. Then, we use this algorithm as a sub-routine to obtain an FPT-AS for the parameter feedback vertex set. Both of these algorithmic results also apply to the more general case of THRESHOLD constraints. We leave it as an interesting open problem to examine if the hardness for treewidth also applies to feedback vertex set, or in the converse direction, if the approximation algorithm for feedback vertex set can be extended to treewidth.

2 Preliminaries

A boolean CSP ψ is defined as a set $\{C_1, \dots, C_m\}$ of m constraints over a set $X(\psi) = \{x_1, \dots, x_n\}$ of n variables and their negations. Each constraint C_i is regarded as a function of literals (positive or negative appearances of variables) mapped to the set $\{0,1\}$, where literals can take the values 0 or 1. Furthermore, we define $|C_j|$ to denote the arity of constraint C_j (the number of literals that occur in C) and $|\psi| = m$ the number of constraints in ψ. For simplicity, we also write $l_i \in C_j$ for a literal l_i and a constraint C_j if l_i appears on C_j.

We will be dealing with boolean constraint satisfaction problems for four well-studied boolean functions: OR constraints, AND constraints, PARITY (or XOR) constraints and MAJORITY constraints. We say that an assignment $t : X \to \{0,1\}$ satisfies a constraint C of type:

- OR, if $\exists l_i \in C$, $t(l_i) = 1$;
- AND, if $\forall l_i \in C$, $t(l_i) = 1$;
- PARITY, if it satisfies some equation $\Sigma_{l_i \in C} t(l_i) = b$ (for $b \in \{0,1\}$) modulo 2;
- MAJORITY, if at least $\lceil |C|/2 \rceil$ literals in C are set to 1. More generally, we may consider THRESHOLD constraints, where a certain threshold number of literals must be set to true to satisfy the constraint.

Let $\text{occ}(\psi) = \sum_{C \in \psi} |C|$ be the total number of variable occurrences in ψ, that is, the total size of the formula. For a variable x, we write ψ_x for the set of all constraints $C \in \psi$

where x occurs either positive or negative; for the functions we consider without loss of generality, no clause contains both literals. Thus, the total number of occurrences of a variable x is equal to $|\psi_x|$.

We are dealing also with MAXCSPs, where given a set of constraints ψ, we are interested in finding an assignment to the variables that maximizes the number of satisfied constraints. The natural decision version of this problem is, given a target k, decide whether there exists an assignment that satisfies at least k constraints. Clearly, the problem where we want to decide whether we can satisfy all the constraints is a special case of the above decision problem since we can set $k = m$, but in some cases we consider this simpler decision version, particularly when we want to show hardness.

In the case of OR constraints, the CSP and MAXCSP problems correspond to the more widely known CNFSAT and MAXCNFSAT problems. In this case we call the constraints *clauses*. When the constraint function is AND, the MAXCSP problem is called MAXDNFSAT. In that case, the constraints are called *terms*.

For a CSP ψ, the incidence graph G^*_ψ is defined as the bipartite graph where we construct one vertex v_i for each variable x_i and one vertex u_j for each constraint C_j and connect v_i with u_j if $x_i \in C_j$.

We are interested in parameterizations of the incidence graph $\mathrm{p}(\mathrm{G}^*_\psi)$ (or simply p^* if G^*_ψ is clear from the context), where p is a structural parameter of G^*_ψ. We are mostly interested in the two most widely studied graph parameters, treewidth tw^* and clique-width cw^*. We refer the reader to standard parameterized complexity textbooks for the definitions, as well as the definitions of standard parameterized complexity terminology used in this paper [8].

3 CNFSAT and MAXCNFSAT

In this section we consider CSPs and MAXCSPs with OR constraints. These problems are widely known as CNFSAT and MAXCNFSAT.

CNFSAT and MAXCNFSAT are very well-studied from the parameterized complexity perspective. It is known that both the decision and the maximization version can be solved exactly in FPT time parameterized by tw^* [1]. However, in the realm of dense graph parameters the best algorithms for CNFSAT and MAXCNFSAT run in XP time parameterized by cw^* [29, 26]. This complexity jump from treewidth to clique-width is known to be unavoidable: it was proven in [24] that CNFSAT is W[1]-hard even when parameterized by incidence modular treewidth mtw^* which is a restriction of cw^*[3].

Below, we attempt to explore two natural ways in order to circumvent this last negative result. The first is to search for a good FPT approximation algorithm, while the second is to consider formulas with an even simpler incidence graph structure.

The first approach is quite fruitful. Indeed, in Section 3.1, we present an FPT approximation scheme for the above parameterizations. Intuitively, the algorithm works as follows: given a formula ϕ with 'small' incidence clique-width, we first examine the formula to see if it contains many or few 'large' clauses. If the formula contains relatively few large clauses, then we simply disregard them. We then know that the incidence graph does not contain 'large' bi-cliques, so by a theorem of Gurski and Wanke [15] the remaining formula has small treewidth and we can solve the problem. If on the other hand the original formula contains

[3] A graph of bounded modular treewidth is a graph of bounded treewidth after merging modules into a single vertex, where a module is a set of vertices with same neighborhood outside of the set).

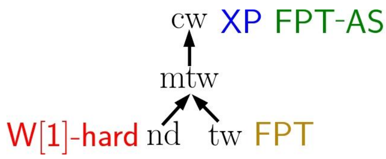


Figure 1 Parameter Map for MAXCNFSAT: algorithmic results hold downwards whereas hardness results hold upwards.

a large number of large clauses, then we observe that we can rely on a random assignment to satisfy almost everything.

The second approach is to consider more restrictive (dense) graph parameters, in the hope that (exact) CNFSAT could be FPT. As mentioned, the reduction of [24] already shows that we have to look at parameters more restrictive than modular treewidth; in fact this reduction constructs a formula whose incidence graph has small feedback vertex set after contracting modules. To this end, in Section 3.2 we concentrate on a much more restricted case: formulas with small incidence *neighborhood diversity* nd^*.

▶ **Definition 1.** A graph $G(V, E)$ has neighborhood diversity $\mathrm{nd}(G) = k$ if we can divide V into k disjoint sets $V_1, \dots V_k$ such that $\forall i \in 1, \dots k, \forall u, v \in V_i \forall w \in V, (w, v) \in E \leftrightarrow (w, u) \in E$.

In other words, $\mathrm{nd}(G) = k$ if V can be divided into k modules that are either cliques or independent sets.

This type of formulas is very restrictive: we only allow a small number of variable- and clause-types, where all same-type variables belong to the same clauses and all same-type clauses involve the same variables. This very restrictive set of formulas is clearly a subset of formulas having $\mathrm{mtw}^* \leq k$, because contracting all modules leaves a graph with *order* at most k (instead of treewidth at most k).

We prove that CNFSAT is W[1]-hard parameterized by nd^*. Intuitively what this result tells us is that, despite the simple structure of the incidence graph, the fact that we can't distinguish between positive and negative appearances of the variables is really responsible for the complexity blow-up.

Figure 1 summarizes known and new results for CNFSAT and MAXCNFSAT for the discussed parameterizations.

3.1 Approximation Algorithm for Clique-width

▶ **Theorem 2.** *There is a randomized algorithm that, given a CNF formula ψ with n variables, m clauses, and incidence clique-width cw, runs in time $f(\epsilon, \mathrm{cw}) \cdot \mathrm{poly}(n + m)$, and outputs a truth assignment that satisfies at least $(1 - \epsilon) \cdot \mathrm{OPT}$ clauses in expectation.*

We formulate the following basic lemma about probability distributions.

▶ **Lemma 3.** *For all $\epsilon, L > 0$ there is a $c = c(\epsilon, L) > 0$ such that all $c' \geq c$ and all sequences $p_1, \dots, p_{c'} \geq 0$ with $\sum_{i=1}^{c'} p_i \leq 1$ have an index $d \leq c/L$ with the property*

$$p_{[d, L \cdot d]} \doteq \sum_{j=d}^{L \cdot d} p_j < \epsilon \,.$$

Proof. Let $\epsilon, L > 0$. We set $c = c(\epsilon, L)$ below. Assume for contradiction that $p_{[d, L \cdot d]} \geq \epsilon$ holds for all $d \in [1, c/L]$. If there are $1/\epsilon + 1$ disjoint intervals $[a_1, L \cdot a_1], \dots, [a_{1/\epsilon+1}, L \cdot a_{1/\epsilon+1}] \subseteq$

$[1, c]$, then we arrive at a contradiction with the fact that the p_i's are non-negative and sum to at most one. Clearly there exists a constant $c = c(\epsilon, L)$ such that $1/\epsilon + 1$ disjoint intervals of the form $[a, La]$ fit into $[1, c]$. This proves the claim. ◀

For an arbitrary given $\epsilon > 0$, we fix $L = \epsilon^{-4}$. We use Lemma 3 as follows: For a CNF formula ψ, we define p_i as the fraction of clauses of size i, that is,

$$p_i \doteq \frac{\left|\left\{ C \in \psi \;\middle|\; |C| = i \right\}\right|}{|\psi|} .$$

Then Lemma 3 gives us a number $d \leq c(\epsilon)$ such that the total fraction of clauses whose size is between d and $\epsilon^{-4}d$ is bounded by ϵ. It is now natural to partition all clauses into short, medium, and long clauses. More precisely, we define $\psi = \psi^{<d} \,\dot\cup\, \psi^{[d,D]} \,\dot\cup\, \psi^{>D}$ for $D = \epsilon^{-4}d$ as follows:

$$\begin{aligned} \psi^{<d} &\doteq \left\{ C \in \psi \;\middle|\; |C| < d \right\}, \\ \psi^{[d,D]} &\doteq \left\{ C \in \psi \;\middle|\; d \leq |C| \leq D \right\}, \text{and} \\ \psi^{>D} &\doteq \left\{ C \in \psi \;\middle|\; |C| > D \right\}. \end{aligned}$$

An immediate corollary to Lemma 3 is thus that we can choose $d \leq c(\epsilon)$ in such a way that $|\psi^{[d,D]}| \leq \epsilon|\psi|$.

▶ **Corollary 4.** *For all $\epsilon > 0$ there is some $c = c(\epsilon) > 0$ such that all CNF formulas ψ have some $d = d(\epsilon) \in [1, c]$ with $|\psi^{[d,\epsilon^{-4}d]}| \leq \epsilon \cdot |\psi|$.*

If $\psi^{[d,D]} = \emptyset$ holds for $D = \epsilon^{-4}d$ and $d \in [1, c(\epsilon)]$, we say that ψ is *ϵ-well separated.* We call ψ *ϵ'-balanced* if, in addition, we have $|\psi^{<d}| \geq \epsilon' m$ and $|\psi^{>D}| \geq \epsilon' m$.

▶ **Lemma 5.** *Let ψ be an ϵ-well separated formula (and thus $V = V(\psi^{<d}) \cup V(\psi^{>D})$).*
Then, for each subset $\hat\psi \subseteq \psi^{>D}$ with $|\hat\psi| > \epsilon^2 m$, there is a variable y such that $|\psi_y^{<d}| \leq \epsilon^2 |\hat\psi_y|$.

That is, for every set $\hat\psi$ that contains a significant fraction of long clauses, there is a variable that occurs $|\hat\psi_y|$ times in $\hat\psi$, but only at most an ϵ^2-fraction of that in the short clauses.

Proof. Let $\hat\psi \subseteq \psi^{>D}$ with $|\hat\psi| > \epsilon^2 m$ Note that the total number of literal occurrences in $\hat\psi$ is $\mathrm{occ}(\hat\psi) > D \cdot \epsilon^2 \cdot m = \epsilon^{-2}dm$. In contrast, $\mathrm{occ}(\psi^{<d}) < dm$. Now suppose that there was no variable y with the claimed properties, that is, suppose that every variable y satisfies $|\psi_y^{<d}| > \epsilon^2|\hat\psi_y|$. Then the total number of variable occurrences in $\psi^{<d}$ can be bounded from below as follows:

$$\mathrm{occ}(\psi^{<d}) = \sum_y |\psi_y^{<d}| > \sum_y \epsilon^2 |\hat\psi_y| = \epsilon^2 \,\mathrm{occ}(\hat\psi) > d \cdot m .$$

This yields a contradiction and thus proves the claim. ◀

Proof of Theorem 2. The algorithm A works as follows. Let $\epsilon' = \epsilon^2$, and we assume w.l.o.g. that $\epsilon < 1/8$. Given a CNF formula ϕ, we compute an ϵ'-well separated formula ψ by dropping all clauses in $\phi^{[d,D]}$. Corollary 4 guarantees that the fraction of deleted clauses is bounded by ϵ'. If ψ is not $\epsilon/2$-balanced, we discard the smaller side (with fewer clauses) and only handle the larger one: If $\psi^{<d}$ is the larger side, we compute an optimal assignment

for $\psi^{<d}$ in FPT time, by using the result of Gurski and Wanke [15]. This way the total number of unsatisfied clauses is at most $\epsilon m/2$, and together with the unsatisfied clauses due to applying Corollary 4, the total number of unsatisfied clauses is smaller than ϵm. Since $\mathrm{OPT} > m/2$, we get the approximation guarantee.

If $\psi^{>D}$ is the larger side, we use a random assignment. This way, at most $\epsilon m/2$ clauses from $\psi^{<d}$ are violated, and in expectation at most a 2^{-D} fraction of clauses from $\psi^{>D}$ are violated. Since 2^{-D} is smaller than $\epsilon/4$, we conclude that – together with unsatisfied clauses due to applying Corollary 4 – at least $(1-\epsilon)m$ clauses are satisfied in expectation.

This finishes the analysis of unbalanced formulas, and in the remaining proof we may assume that ψ is $\epsilon/2$-balanced. To handle this case, we determine a set of variables Y such that

- there are at most $\epsilon m/4$ short clauses with variables from Y and
- there are at most $\epsilon^2 m$ long clauses that contain $\leq 1/\epsilon$ variables from Y.

Before we construct Y, let us verify that the properties of Y imply the correctness of the theorem. Our algorithm computes a satisfying assignment of the short clauses without variables from Y, again using the result of Gurski and Wanke [15]. Afterwards it assigns values uniformly at random to the variables in Y.

There are at most $\epsilon' m = \epsilon^2 m$ unsatisfied clauses due to applying Corollary 4, $\epsilon m/4$ short clauses clauses that we did not consider when satisfying clauses from $\psi^{<d}$, and $\epsilon^2 m$ clauses from $\psi^{>D}$ that we did not consider in the random assignment. Additionally, in expectation there are less than $2^{-1/\epsilon}m$ clauses left unsatisfied from the remaining $|\psi^{>D}| - \epsilon^2 m$ clauses from $\psi^{>D}$. Since we assumed that $\epsilon < 1/8$, the theorem follows.

To construct the set Y, we iteratively apply Lemma 5 with the parameter $\epsilon/4$. Initially, we set $\hat{\psi} = \psi^{>D}$. In each iteration, we identify a variable y according to the lemma and add the variable to Y. In the subsequent iterations, we mark y to be inactive and handle it as if it was not contained in any clause. Whenever we identify a clause C that has at least $1/\epsilon$ inactive variables (i. e., variables from Y), we remove C from $\hat{\psi}$. We continue this process until $|\hat{\psi}| \leq \epsilon^2$. Note that applying Lemma 5 for $\epsilon/4$ but having an ϵ'-well separated formula ensures that at all times, all clauses in $\hat{\psi}$ have sufficiently many literals to apply Lemma 5. Therefore the process terminates and the generated set Y has the aimed-for properties since $|Y| \leq m/\epsilon$. ◀

3.2 Hardness for Neighborhood Diversity

▶ **Theorem 6.** CNFSAT *parameterized by the incidence neighborhood diversity* nd* *is* W[1]*-hard.*

Proof. The reduction is from k-Multicolored Clique. Given a k-partite graph $G(V_1,\ldots,V_k,E)$, where $|V_i| = n$, we construct a formula ϕ as described below.

First, we construct k groups of variables $X = X_1 \cup X_2, \ldots \cup X_n$, with $|X_i| = \log n$. Variable x_i^j should correspond to the i^{th} bit in the binary representation of the vertices of group V_j. We also construct $\binom{k}{2}$ groups of clauses as follows: for each non-edge (u,v) between vertex parts V_i and V_j, we create a clause that represents u and v in binary, with a positive literal representing the bit 0 and a negative one the bit 1. For example, if $u_3^i \in V_i$, $u_1^j \in V_j$, and $\left(u_3^i, u_1^j\right) \notin E$, then we create the clause $\left(\neg x_0^i \vee \neg x_1^i \vee x_0^j \vee \neg x_1^j\right)$. Each clause should contain $2\log n$ literals.

The intuition behind the construction is that, for every pair i,j, there is a bijection between pairs of vertices $\left(v_r^i, v_s^j\right)$ in G_ϕ^* and clauses that consist of variables $X_i \cup X_j$ such that:

$$(v_r^i, v_s^j) \leftrightarrow \left(\bigvee_{\substack{q:qth \text{ bit} \\ \text{of } r_{(2)} \text{ is } 0}} x_q^i \bigvee_{\substack{q:qth \text{ bit} \\ \text{of } r_{(2)} \text{ is } 1}} \neg x_q^i \bigvee_{\substack{q:qth \text{ bit} \\ \text{of } s_{(2)} \text{ is } 0}} x_q^j \bigvee_{\substack{q:qth \text{ bit} \\ \text{of } s_{(2)} \text{ is } 1}} \neg x_q^j \right)$$

Indeed, for any pair X_i, X_j, there are $2^{2\log n} = n^2$ different ways to create a clause using all the variables from $X_i \cup X_j$. Furthermore, by construction, every non-edge of G corresponds to a unique clause in ϕ. Thus every edge in G corresponds to a clause not in ϕ and vice versa.

Having that said, given a satisfying assignment, for every pair of variable groups, the unique unsatisfiable clause c (variables assigned value 0 appear positive whereas those assigned value 1 appear negative) does not belong in ϕ (since all clauses of ϕ are satisfied). This clause corresponds to an edge in G.

On the other hand, given a clique in G, any pair of clique vertices corresponds to some clause not in ϕ. Thus, the assignment that falsifies this clause should satisfy all other clauses of this given pair of variables.

Notice that the neighborhood diversity of the incidence graph of the constructed formula is $k + \binom{k}{2}$ (one module for every variable or clause group).

◀

4 From Treewidth to Clique-width

Perhaps the main conclusion of the previous section is that the move from sparse graph parameters (such as treewidth) to dense graph parameters can significantly change the complexity of a CSP. CNFSAT becomes W-hard even for an extremely restricted dense graph parameter (neighborhood diversity), but it at least remains well-approximable even in the much more general case of clique-width.

In this section we observe that the transition from sparse to dense parameters can have wildly varying effects for different CSPs. First, by modifying our reduction for CNFSAT we show that the problem of maximizing the number of satisfied AND constraints is W[1]-hard parameterized by neighborhood diversity. Furthermore, because the maximum number of constraints that could be satisfied in our reduction is also bounded by some function of the parameter, we show that if an approximation algorithm with performance similar to that for MAXCNFSAT could be devised for this problem then FPT=W[1]. Thus, MAXDNFSAT, which is FPT parameterized by treewidth, becomes significantly harder for dense parameters.

▶ **Theorem 7.** *Suppose that there exists an* FPT-AS *which* $\forall\epsilon > 0$ *computes an* $(1-\epsilon)$*-approximate solution for* MAXDNFSAT *and runs in time* $f(\mathrm{nd}^*, \epsilon) \cdot \mathrm{poly}(n)$*, where* nd^* *is the incidence neighborhood diversity of an* AND *CSP* ϕ*. Then,* $\mathsf{FPT} = \mathsf{W}[1]$*.*

We can observe however that, unlike MAXCNFSAT and MAXDNFSAT, there exist natural CSPs for which the transition from treewidth to clique-width does not entail any significant change in complexity. A case in point is MAXPARITY. Recall that finding an assignment that satisfies the maximum number of linear equations modulo 2 is one of the prototypical APX-hard problems [17], despite the fact that deciding if an assignment can satisfy all equations is in P (by Gauss elimination). Here, we show that MAXPARITY is FPT, regardless of whether the parameter is treewidth or clique-width. The main intuition is that in this CSP, negations are (almost) irrelevant. Thus, unlike the case of CNFSAT and

MAXDNFSAT the incidence graph captures much more of the real structure of the CSP instance.

▶ **Theorem 8.** *Given an input* PARITY CSP *instance* ϕ*,* MAXPARITY *can be solved optimally in time* $f(\mathrm{cw}^*)|\phi|^{O(1)}$*, where* cw^* *is the incidence clique-width of* ϕ*.*

5 Below treewidth: The case of Majority

In this section we deal with CSPs where each constraint is a MAJORITY or a THRESHOLD constraint. In this problem each constraint is supplied with an integer value t (the threshold) and it is satisfied if and only if at least t of its literals are set to true. MAJORITY is the special case of this predicate where t is always equal to half the arity of each constraint.

MAJORITY and THRESHOLD constraints are of course some of the most natural and well-studied predicates in many contexts: for example, MAXCSP for such constraints contains the complexity of finding an assignment that satisfies as many inequalities as possible in a 0-1 Integer Linear Program whose coefficients are in $\{-1, 0, 1\}$. This problem, sometimes called MAXIMUM FEASIBLE SUBSET has been well-studied in the literature [9, 3, 2]. MAJORITY constraints also play a central role in learning theory [10, 16] and in hardness of approximation [6].

5.1 Hardness

We first consider the complexity of deciding whether a CSP with THRESHOLD constraints can be completely satisfied parameterized by the treewidth of the incidence graph. Unfortunately, this turns out to be a hard problem, even for the special case of MAJORITY constraints.

▶ **Theorem 9.** MAJORITY *parameterized by the incidence treewidth* tw^* *is* W[1]*-hard.*

The full proof of this theorem is given in the appendix.

5.2 Exact Algorithms

Motivated by the negative result of Theorem 9 we now investigate the complexity of MAJORITY for more restricted parameters. The first parameter we consider is the vertex cover of the incidence graph. This is a natural, though quite restrictive, parameter which is often considered for problems which are W-hard for treewidth. Note that parameterizing by incidence vertex cover is almost equivalent (though slightly more general) to parameterizing by the number of constraints.

In the following theorem we establish that for this parameter even the maximization problem for the (more general) THRESHOLD constraints is FPT.

▶ **Theorem 10.** MAXTHRESHOLD *parameterized by the number of constraints is* FPT*.*

Proof. Let ϕ be a CSP with $m = |\phi|$ constraints. First, we will exhaustively search among all possible 2^m subsets of ϕ for a feasible solution of maximum size. For each such candidate S we need to verify whether there is an assignment that satisfies all $C \in S$.

Now, given a candidate set of constraints S where $|S| = k \leq m$, our goal is to construct an equivalent ILP π_S over a variable-set L, where $|L|$ is bounded by a function of m. We then know that we can solve π_S in FPT time [21].

We divide the set of variables X appearing in S into at most 3^k disjoint sets X_i according to the state of a variable in regards to each of the k constraints of S: appearing positive

(0), appearing negative (1), not appearing (2). Now, for each set X_i we define a variable of the ILP l_i measuring how many variables from X_i are set to true. Now we can construct the following equivalent ILP: for each $C_j \in S$ we create an inequality involving those X_is that participate in C_j. The equation is going to be $\sum l_i^+ - \sum l_i^- \geq t(C_j) - \sum |X_i^-|$, where X_i^+ (resp. X_i^-) are all the sets of variables that appear positive (resp. negative) in C_j. Furthermore we have the constraints that $|X_i| \geq l_i \geq 0$. Finding a feasible integer solution for π_S gives a feasible assignment for S.

◀

▶ **Corollary 11.** THRESHOLD *parameterized by the incidence vertex cover* vc* *is* FPT.

Proof. Given a threshold CSP ϕ with a variable set X and a vertex cover S with size k of the incidence graph, we define $S_X, S_Y \subseteq S$ to be vertices of S corresponding to variables and constraints respectively. Since $|S_X| \leq k$ we can consider all possible assignments of S_X. For each such assignment we can compute the exact number of satisfied constraints corresponding to $\phi \setminus S_Y$. Thus we need to solve at most 2^k instances with bounded number of constraints.

◀

5.3 Approximation

The results of Theorem 9 naturally pose the following question: can we evade the W-hardness of MAJORITY by designing an FPT-AS for the problem? In this section, though we do not resolve this question, we give some first positive indication that this may be possible. We consider THRESHOLD parameterized by the incidence graph's feedback vertex set (that is, the number of vertices that need to be deleted to make the graph acyclic). This is a natural, well-studied parameter that generalizes vertex cover but is a restriction of treewidth. It is also connected to the concept of back-door sets to acyclicity, which is well-studied in the parameterized CSP literature [23, 12].

Observe that approximating this CSP is non-trivial, since MAJORITY with constraints of arity two already generalizes Max-2SAT, and is hence APX-hard. On the other hand, MAXMAJORITY can easily be 2-approximated by considering any assignment and its negation. Hence, the natural goal here is a $(1-\epsilon)$ approximation ratio. Using Corollary 11 as a sub-routine we can give an FPT-AS scheme which achieves exactly this.

▶ **Theorem 12.** *There exists an approximation scheme for* THRESHOLD *such that* $\forall \epsilon > 0$ *and any input* ϕ *it runs in time* $f(k,\epsilon)|\phi|^{O(1)}$ *with* $k = \text{fvs}^*$ *being the incidence feedback vertex set of the incidence graph of* ϕ*, and returns an* $(1-\epsilon)$ *approximate solution.*

Proof. We consider two cases:

- If $|\phi| \leq {}^{2k}/_{\epsilon}$, this case reduces to Theorem 10 because the graph has vertex cover bounded by a function of k, ϵ.
- If $|\phi| > {}^{2k}/_{\epsilon}$, then with similar argumentation as Corollary 10 we can consider that the fvs(G^*_ϕ) contains only constraint vertices (that is, we guess the assignment of variables in the feedback vertex set). We now proceed by simply deleting these constraints from the instance. Since the resulting instance is acyclic we can find the optimal solution in the remaining constraints of ϕ in polynomial time. Call the produced solution SOL and the optimal solution OPT. Then we have: SOL $\geq$ OPT $- k$. Now, observe that OPT $\geq |\phi|/2$ because if an assignment satisfies less than half of the constraints, its negation satisfies the rest. Therefore OPT $> {}^{k}/_{\epsilon}$ which gives $\frac{\text{OPT}-k}{\text{OPT}} > 1-\epsilon$.

◀

References

1 M. Alekhnovich and A. A. Razborov. Satisfiability, branch-width and tseitin tautologies. *Computational Complexity*, 20(4):649–678, 2011.

2 E. Amaldi and V. Kann. The complexity and approximability of finding maximum feasible subsystems of linear relations. *Theor. Comput. Sci.*, 147(1&2):181–210, 1995.

3 E. Amaldi and V. Kann. On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. *Theor. Comput. Sci.*, 209(1-2):237–260, 1998.

4 P. Austrin and S. Khot. A characterization of approximation resistance for even k-partite csps. In R. D. Kleinberg, editor, *Innovations in Theoretical Computer Science, ITCS '13, Berkeley, CA, USA, January 9-12, 2013*, pages 187–196. ACM, 2013.

5 B. Courcelle, J. A. Makowsky, and U. Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory Comput. Syst.*, 33(2):125–150, 2000.

6 A. De, E. Mossel, and J. Neeman. Majority is stablest: discrete and sos. In D. Boneh, T. Roughgarden, and J. Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 477–486. ACM, 2013.

7 M. Dom, D. Lokshtanov, S. Saurabh, and Y. Villanger. Capacitated domination and covering: A parameterized perspective. In *IWPEC*, pages 78–90. Springer, 2008.

8 R. G. Downey and M. R. Fellows. *Fundamentals of Parameterized Complexity.* Texts in Computer Science. Springer, 2013.

9 K. M. Elbassioni, R. Raman, S. Ray, and R. Sitters. On the approximability of the maximum feasible subsystem problem with 0/1-coefficients. In C. Mathieu, editor, *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2009, New York, NY, USA, January 4-6, 2009*, pages 1210–1219. SIAM, 2009.

10 V. Feldman, V. Guruswami, P. Raghavendra, and Y. Wu. Agnostic learning of monomials by halfspaces is hard. *SIAM J. Comput.*, 41(6):1558–1590, 2012.

11 S. Gaspers and S. Szeider. Kernels for global constraints. In T. Walsh, editor, *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, pages 540–545. IJCAI/AAAI, 2011.

12 S. Gaspers and S. Szeider. Backdoors to acyclic SAT. In A. Czumaj, K. Mehlhorn, A. M. Pitts, and R. Wattenhofer, editors, *Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part I*, volume 7391 of *Lecture Notes in Computer Science*, pages 363–374. Springer, 2012.

13 S. Gaspers and S. Szeider. Guarantees and limits of preprocessing in constraint satisfaction and reasoning. *Artif. Intell.*, 216:1–19, 2014.

14 M. Grohe. The structure of tractable constraint satisfaction problems. In R. Kralovic and P. Urzyczyn, editors, *MFCS 2006, Stará Lesná, Slovakia, August 28-September 1, 2006, Proceedings*, volume 4162 of *Lecture Notes in Computer Science*, pages 58–72. Springer, 2006.

15 F. Gurski and E. Wanke. The tree-width of clique-width bounded graphs without $K_{n,\ n}$. In U. Brandes and D. Wagner, editors, *Graph-Theoretic Concepts in Computer Science, 26th International Workshop, WG 2000, Konstanz, Germany, June 15-17, 2000, Proceedings*, volume 1928 of *Lecture Notes in Computer Science*, pages 196–205. Springer, 2000.

16 V. Guruswami and P. Raghavendra. Hardness of learning halfspaces with noise. *SIAM J. Comput.*, 39(2):742–765, 2009.

17 J. Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001.

18 S. Khanna, M. Sudan, and D. P. Williamson. A complete classification of the approximability of maximization problems derived from boolean constraint satisfaction. In F. T. Leighton and P. W. Shor, editors, *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing, El Paso, Texas, USA, May 4-6, 1997*, pages 11–20. ACM, 1997.

19 S. Khot and R. Saket. Approximating csps using LP relaxation. In M. M. Halldórsson, K. Iwama, N. Kobayashi, and B. Speckmann, editors, *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part I*, volume 9134 of *Lecture Notes in Computer Science*, pages 822–833. Springer, 2015.

20 M. Lampis. Algorithmic meta-theorems for restrictions of treewidth. *Algorithmica*, 64(1):19–37, 2012.

21 H. W. Lenstra Jr. Integer programming with a fixed number of variables. *Mathematics of operations research*, 8(4):538–548, 1983.

22 D. Marx. Parameterized complexity and approximation algorithms. *Comput. J.*, 51(1):60–78, 2008.

23 S. Ordyniak, D. Paulusma, and S. Szeider. Satisfiability of acyclic and almost acyclic CNF formulas. *Theor. Comput. Sci.*, 481:85–99, 2013.

24 D. Paulusma, F. Slivovsky, and S. Szeider. Model counting for CNF formulas of bounded modular treewidth. In N. Portier and T. Wilke, editors, *STACS 2013, February 27 - March 2, 2013, Kiel, Germany*, volume 20 of *LIPIcs*, pages 55–66. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2013.

25 R. Pichler, S. Rümmele, S. Szeider, and S. Woltran. Tractable answer-set programming with weight constraints: bounded treewidth is not enough. *TPLP*, 14(2):141–164, 2014.

26 S. H. Sæther, J. A. Telle, and M. Vatshelle. Solving maxsat and #sat on structured CNF formulas. In C. Sinz and U. Egly, editors, *SAT 2014 - Vienna, Austria, July 14-17, 2014. Proceedings*, volume 8561 of *Lecture Notes in Computer Science*, pages 16–31. Springer, 2014.

27 M. Samer and S. Szeider. Constraint satisfaction with bounded treewidth revisited. *J. Comput. Syst. Sci.*, 76(2):103–114, 2010.

28 T. J. Schaefer. The complexity of satisfiability problems. In R. J. Lipton, W. A. Burkhard, W. J. Savitch, E. P. Friedman, and A. V. Aho, editors, *Proceedings of the 10th Annual ACM Symposium on Theory of Computing, May 1-3, 1978, San Diego, California, USA*, pages 216–226. ACM, 1978.

29 F. Slivovsky and S. Szeider. Model counting for formulas of bounded clique-width. In L. Cai, S. Cheng, and T. W. Lam, editors, *ISAAC 2013, Hong Kong, China, Proceedings*, volume 8283 of *Lecture Notes in Computer Science*, pages 677–687. Springer, 2013.

30 S. Szeider. Not so easy problems for tree decomposable graphs. *CoRR*, abs/1107.1177, 2011.

31 S. Szeider. The parameterized complexity of constraint satisfaction and reasoning. In H. Tompits, S. Abreu, J. Oetsch, J. Pührer, D. Seipel, M. Umeda, and A. Wolf, editors, *INAP 2011, and WLP 2011, Vienna, Austria, September 28-30, 2011, Revised Selected Papers*, volume 7773 of *Lecture Notes in Computer Science*, pages 27–37. Springer, 2011.

32 S. Szeider. The parameterized complexity of k-flip local search for SAT and MAX SAT. *Discrete Optimization*, 8(1):139–145, 2011.

33 L. Trevisan. Inapproximability of combinatorial optimization problems. *Electronic Colloquium on Computational Complexity (ECCC)*, 2004.

6 Missing Proofs

6.1 Proof of Theorem 7

Proof of Theorem 7. The reduction is again from k-Multicolored Clique and follows similar ideas with that of Theorem 6.

Again, given a k-partite graph $G(V_1, V_2, \ldots, V_k, E)$, where $|V_i| = n$, we construct k groups of variables as in Theorem 6 and $\binom{k}{2}$ groups of AND constraints but this time we create a constraint that represents u and v in binary for each (existing) edge (u, v) between vertex parts V_i and V_j. The constraint representing edge (u, v) is satisfied if and only if the truth assignment corresponds to the binary representations of u and v. There will be $|E|$ constraints in total and each constraint will contain $2 \log n$ literals.

A selection of $v_i \in V_i$ corresponds to an assignment over the variables in X_i. Given two variable groups X_i, X_j and an assignment over these variables, at most one constraint can be satisfiable. Intuitively, this constraint (if it exists) corresponds to an edge between V_i, V_j. Finding the assignment that satisfies the most AND constraints is equivalent with finding the k-densest subgraph in the k-partite graph. We shall prove that having an FPT-AS for the MAXDNFSAT problem implies that we can solve k-Multicolored Clique in FPT time.

Suppose that we have an FPT-AS for MAXDNFSAT such that for all $\epsilon > 0$, it computes an $(1 - \epsilon)$ approximate solution and runs in time $f(k, \epsilon) \cdot \text{poly}(n)$.

Set $\frac{1}{\epsilon} = k^2 (> \binom{k}{2})$. The running time of this algorithm is $f(k, \epsilon) \cdot \text{poly}(n) = g(k) \cdot \text{poly}(n)$.

- If G has a k-clique, then there should be $\binom{k}{2}$ satisfied constraints, so the algorithm shall return a value $\geq (1 - \epsilon)\binom{k}{2}$, which rounds up to $\binom{k}{2}$.
- If G has no k-clique, then there are at most $\binom{k}{2} - 1$ satisfied constraints, which is the best value the algorithm can return in this case.

Thus, by running the approximation algorithm for MAXDNFSAT for $\epsilon = \frac{1}{k^2}$ we should be able to distinguish between having a k clique and a clique of lesser size in FPT time.

◀

6.2 Proof of Theorem 8

Proof of Theorem 8. We rely on the meta-theorem of [5] which states that all problems expressible in (an optimization version of) CMSO_1 can be solved in linear time. We recall that in this context we are allowed to express problems that quantify over sets of vertices of the input graph, and also express constraints on the cardinalities of these sets modulo a constant number.

The key intuition here is that, when given a linear equation mod 2 of the form $\sum_i l_i = b$ where $b \in \{0, 1\}$ and the l_i are literals, we may view it equivalently as a constraint of the form $\sum_i x_i = b'$, where all the x_i are variables and $b' = b$ if and only if the original constraint contained an even number of negated literals on the left hand side.

Having performed the above pre-processing we can now express our problem in CMSO_1. We are looking for the largest set of constraints S (which are vertices of the incidence graph), such that there exists a set of variables X which satisfies the following: every constraint in S has a number of neighbors in X that is equal to its right-hand side (mod 2). Thus, S is the set of satisfied constraints and X the set of variables set to 1 in the satisfying assignment.

◀

6.3 Proof of Theorem 9

Proof of Theorem 9. We first prove that a generalization of MAJORITY, THRESHOLD, is W[1]-hard parameterized by tw*. For THRESHOLD, the constraints are each assigned a threshold function $t : C \to \mathbb{N}$ and in order for a constraint C to be satisfied, at least $t(C)$ literals need to be set to true. The question is again whether we can satisfy all the constraints. We then show how to reduce this more general version to MAJORITY parameterized by tw*, where essentially, for each constraint C, $t(C) = \lceil |C|/2 \rceil$.

The reduction is from k-Capacitated Vertex Cover parameterized by the treewidth of the graph. The definition of this problem is given in Def. 13. This problem was shown to be W[1]-hard parameterized by treewidth in [7].

▶ **Definition 13.** A graph with capacities $G(V, E, w)$ is a graph supplemented with a function on the vertices $w : V \to \mathbb{N}$ (indicating the capacity of each vertex), where $0 \leq w(v) \leq d(v)$, with $d(v)$ being the degree of a vertex v. A set $S \subseteq V$ is a *capacitated vertex cover* of G if there exists a function $g : E \to S$ that assigns for each edge $e = (u, v)$ a guarding endpoint u or v in S and, furthermore, the total number of edges guarded by any vertex v does not exceed $w(v)$.

k-Capacitated Vertex Cover: Given a graph with capacities $G(V, E, w)$ and an integer $k \leq n$, decide whether there exists a capacitated vertex cover of size at most k.

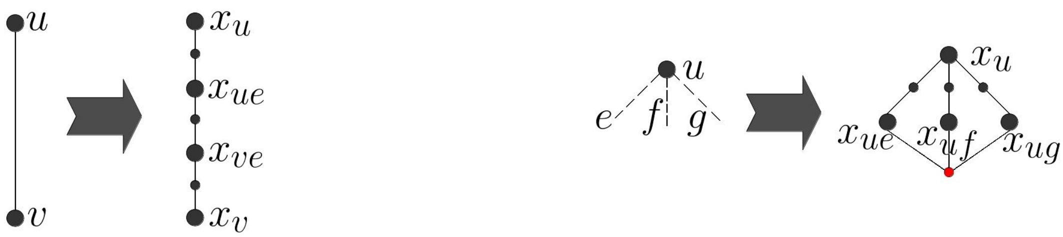
Given a graph with capacities $G(V, E, w)$, we construct a THRESHOLD CSP ϕ as follows. We first construct the following variables: for every $v \in V$ we construct a variable x_v which should be true if v was chosen in the capacitated vertex cover; also, for every $e = (u, v) \in E$ we construct two variables x_{eu}, x_{ev} (x_{eu} being true indicates that u guards e).

Now we need to express the following statements with threshold constraints.

1. *"If a vertex v guards an edge e then v is in the vertex cover"*: for all $e \in E$ and each of its incident vertices we construct one constraint $(\neg x_{ve}, x_v)$ with threshold 1;
2. *"Every edge is covered"*: for all $e = (u, v) \in E$, we construct one constraint (x_{ue}, x_{ve}) with threshold 1.
3. *"No vertex exceeds its capacity"*: for all $v \in V$, we construct one constraint of size $d(u)$ that contains literals $\neg x_{ve_i}$ for every incident edge e_i of v. This constraint should have threshold $d(v) - w(v)$.
4. *"At most k vertices were picked in the vertex cover"*: we construct one constraint which should contain all literals $\neg x_{v_j}$ for every vertex $v_j \in V$ with threshold $|V| - k$.

The forward direction is clear: given a capacitated vertex cover S of size at most k and a valid function $g : E \to S$, setting variables to true as suggested above should satisfy all threshold constraints. For the opposite direction, if all the constraints have been satisfied, then at most k variables x_v should be set to true from the last constraint. Vertices corresponding to these variables are put in the vertex cover. The values of $g(e)$ are set according to the values of variables x_{ue}, x_{ve}: at least one of the two should be true, according to constraint 2. If x_{ue} is true then set $g(e) = u$. If both x_{ue}, x_{ve} are true, then set $g(e)$ arbitrarily. Constraints 1 and 2 should certify that the selected set S is indeed a capacitated vertex cover, whereas constraint 3 verifies that capacities are met.

Last we need to show that the incidence treewidth $\mathrm{tw}(\mathrm{G}^*_\phi)$ of ϕ is bounded by a function of the treewidth of the original graph $\mathrm{tw}(G)$. Indeed, G^*_ϕ is a subdivision of the original graph G with the addition of a false twin vertex for every vertex in G and a universal vertex connected to all original vertices. To see this, first observe that for every edge $e = (u, v)$ in G we shall construct four variable vertices x_u, x_{ue}, x_{ve}, x_v in G^*_ϕ, where consecutive vertices in this sequence are connected through constraint vertices (the first and last pair because of 1 and the middle pair because of 2, see figure Fig. 2a). Subdividing an edge (u, v) to $(u, w), (w, v)$ doesn't increase the treewidth of the graph: create a new bag containing all u, v, w and connect it to some bag which contains both u and v. Then for every vertex v in G whose incident edges are $e_1, \ldots, e_{d(v)}$, the constraint vertex from 3 which connects to vertices $x_{ve_1}, \ldots, x_{ve_{d(v)}}$ can be considered as a false twin of x_v in G_ϕ (if we ignore the fact that the edges (x_v, x_{ve_i}) are subdivided, see figure Fig. 2b). Creating a false twin u' for every original vertex u at most doubles the treewidth: for every bag that u appears put u' as well. Last, because of the existence of constraint 4 we have one vertex corresponding to this constraint which connects to all x_{v_j}, $v_j \in V$. The addition of one vertex increases the treewidth by at most one (put the new vertex in all bags). So $\mathrm{tw}(\mathrm{G}^*_\phi) \leq 2 \cdot \mathrm{tw}(G) + 1$



(a) Subdivision gadget   **(b)** Twin gadget

Figure 2 Gadgets in the new graph G_ϕ

In order to show now that MAJORITY is $W[1]$-hard starting from THRESHOLD, there are two cases we need to resolve: when a threshold is lower than the majority and when it is higher.

When a constraint C has threshold $t(C) < \lceil |C|/2 \rceil$, we construct a new constraint C' by adding d dummy variables to it such that the new constraint has $t(C') = \lceil |C'|/2 \rceil$. A dummy variable can always be satisfied by setting it to true, so $t(C') = t(C) + d$. So overall in order to figure out d, we need to solve the equation $t(C) + d = \lceil |C| + d/2 \rceil$, which gives $d = |C| - 2t(C)$. Observe that adding new variables to a constraint corresponds to adding leaves to the constraint vertex in the incidence graph, which does not increase the treewidth.

On the other hand, if a constraint C has threshold $t(C) > \lceil |C|/2 \rceil$, again we add d' new variables but this time we make sure that these variables cannot be set to true. In order to certify this, we should also add for each new variable x one additional constraint $(\neg x)$. In this second case, $t(C') = t(C)$, so the number of additional variables $d' = 2t(C) - |C|$. Again, the incidence treewidth does not change since this time we simply append P_2s on the constraint vertex.

◀