Proceedings Paper:
Jain, P., Kanesh, L., Panolan, F. orcid.org/0000-0001-6213-8687 et al. (4 more authors) (2024) Max-SAT with Cardinality Constraint Parameterized by the Number of Clauses. In: Soto, J.A. and Wiese, A., (eds.) Proceedings of Latin American Theoretical Informatics (LATIN 2024). Latin American Theoretical Informatics (LATIN 2024), 18-22 Mar 2024, Puerto Varas, Chile. Lecture Notes in Computer Science . Springer Nature , pp. 223-237. ISBN 978-3-031-55601-2

https://doi.org/10.1007/978-3-031-55601-2_15

Max-SAT with Cardinality Constraint Parameterized by the Number of Clauses

Pallavi Jain[1], Lawqueen Kanesh[1], Fahad Panolan[2], Souvik Saha[3][0000−0001−8322−0639], Abhishek Sahu[4], Saket Saurabh[3,5], and Anannya Upasana[3]

[1] Indian Institute of Technology Jodhpur, India {pallavi,lawqueen}@iitj.ac.in
[2] School of Computing, University of Leeds,UK
f.panolan@leeds.ac.uk
[3] The Institute of Mathematical Sciences, HBNI , India
{souviks,saket,anannyaupas}@imsc.res.in
[4] National Institute of Science Education and Research, HBNI, India
abhisheksahu@niser.ac.in
[5] The Institute of Mathematical Sciences, HBNI , India
[6] University of Bergen, Norway

Abstract. Max-SAT with cardinality constraint (CC-Max-SAT) is one of the classical NP-complete problems. In this problem, given a CNF-formula Φ on n variables, positive integers k, t, the goal is to find an assignment β with at most k variables set to true (also called a weight k-assignment) such that the number of clauses satisfied by β is at least t. The problem is known to be W[2]-hard with respect to the parameter k. In this paper, we study the problem with respect to the parameter t. The special case of CC-Max-SAT, when all the clauses contain only positive literals (known as Maximum Coverage), is known to admit a $2^{\mathcal{O}(t)}n^{\mathcal{O}(1)}$ algorithm. We present a $2^{\mathcal{O}(t)}n^{\mathcal{O}(1)}$ algorithm for the general case, CC-Max-SAT. We further study the problem through the lens of kernelization. Since Maximum Coverage does not admit polynomial kernel with respect to the parameter t, we focus our study on $K_{d,d}$-free formulas (that is, the clause-variable incidence bipartite graph of the formula that excludes $K_{d,d}$ as a subgraph). Recently, in [Jain et al., SODA 2023], an $\mathcal{O}(dt^{d+1})$ kernel has been designed for the Maximum Coverage problem on $K_{d,d}$-free incidence graphs. We extend this result to Max-SAT on $K_{d,d}$-free formulas and design a $\mathcal{O}(d4^{d^2}t^{d+1})$ kernel.

Keywords: FPT · Kernel · Max-SAT.

1 Introduction

SAT, the first problem that was shown to be NP-complete, is one of the most fundamental, important, and well-studied problems in computer science. In this problem, given a CNF-formula Φ, the goal is to decide if there exists an assignment β that sets the variables of the given formula to true or false such that the formula is satisfied. The optimisation version of the problem is Max-SAT, in which the goal is to find an assignment that satisfies the maximum number of clauses. There are several applications of Max-SAT e.g., cancer therapy design, resource allocation, formal verification, and many more [15,10]. In this paper, we study the Max-SAT problem under cardinality constraint, known as CC-Max-SAT, which is a generalisation of Max-SAT. In this problem, given a CNF-formula Φ, positive integers k and t; the goal is to find an assignment β with at most k variables set to true (also called a weight k-assignment) such that the number of clauses satisfied by β is at least t.

CC-Max-SAT has been studied from the approximation viewpoint. It admits a $(1-\frac{1}{e})$-factor approximation algorithm [13]. Here, e is the base of the natural logarithm. Feige [5] showed that this approximation algorithm is optimal even for a special case of CC-Max-SAT, where the clauses contain only positive literals. This special case of CC-Max-SAT is known as Maximum Coverage, in which we are given a family of subsets, $\mathcal{F}$, over a universe U, and the goal is to find a subfamily $\mathcal{F}' \subseteq \mathcal{F}$ of size at most k such that the number of elements of U that belong to a set in $\mathcal{F}'$ is at least t.

In this paper, we study the problem in the realm of parameterized complexity. The Maximum Coverage problem is well-studied from the parameterized viewpoint and is known to be $\mathsf{W}[2]$-hard with respect to the parameter k [3]. In fact, it is known that assuming $\mathsf{GAP\text{-}ETH}$, we cannot hope for an approximation algorithm with factor $(1-\frac{1}{e}+\epsilon)$ in running time $f(k,\epsilon)n^{\mathcal{O}(1)}$ for Maximum Coverage [9,12]. Thus, the next natural parameter to study is t (the minimum number of clauses to be satisfied).

As early as 2003, Bläser [2] showed that Maximum Coverage admits a $2^{\mathcal{O}(t)}n^{\mathcal{O}(1)}$ algorithm. We first generalise this result to CC-Max-SAT.

Theorem 1. *There exists a deterministic algorithm that solves the* CC-Max-SAT *problem in time* $2^{\mathcal{O}(t)}n\log n$.

The next natural question is, "Does the problem admit a polynomial kernel with respect to the parameter t?". That is, does there exist a polynomial time algorithm that, given an instance (Φ,k,t) of CC-Max-SAT returns an equivalent instance (Φ',k',t') of CC-Max-SAT whose size is bounded by a polynomial in t. Unfortunately, Maximum Coverage does not admit a polynomial kernel with respect to the parameter t unless $\mathsf{PH} = \Sigma_{\mathsf{p}}^{3}$ [4]. Thus, we cannot hope for a polynomial kernel with respect to the parameter t for CC-Max-SAT. So a natural question is for which families of input does the problem admit a polynomial kernel.

In 2018, Agrawal et al. [1] designed a kernel for a special case of Maximum Coverage when every element of the universe appears in at most d sets. In this kernel, the universe is bounded by $\mathcal{O}(dt^2)$, and the size of the family is bounded by $\mathcal{O}(dt)$. Recently, in 2023, Jain et al. [7] designed an $\mathcal{O}(dt^{d+1})$ kernel for the Maximum Coverage problem on $K_{d,d}$-free incident graphs. Here, by an incident graph, we mean a bipartite graph $G=(A\cup B,E)$ where A contains a vertex for each element of the set family and B contains a vertex for each element of the ground set and there is an edge between $u\in A$ and $v\in B$ if the

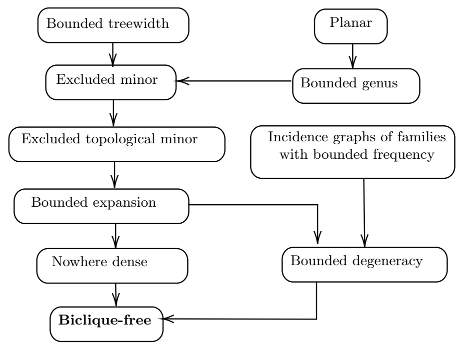

Fig. 1. Inclusion relation between the class of biclique-free graphs considered in this paper and well-studied graph classes in the literature (figure based on [14]). If there is an arrow of the form $A \to B$, then class A is a subclass of B.

set corresponding to u contains the element corresponding to v. $K_{d,d}$ denotes a complete bipartite graph with bipartitions of size d each. $K_{d,d}$-free (also called biclique-free) graphs are very extensive classes of graphs and include many well-studied graph classes such as bounded treewidth graphs, graphs that exclude a fixed minor, graphs of bounded expansion, nowhere-dense graphs, and graphs of bounded degeneracy. That is, for any of the classes – bounded treewidth graphs, graphs that exclude a fixed minor, graphs of bounded expansion, nowhere-dense graphs, and graphs of bounded degeneracy, there is a p such that the class is contained in the class of $K_{p,p}$-free graphs (see Figure 1 for an illustration of the inclusion relation between these classes). For a CNF-formula Φ, let G_Φ denote the clause-variable incident bipartite graph of Φ. That is, the vertex set of G_Φ is $\mathsf{var}(\Phi) \uplus \mathsf{cla}(\Phi)$. For each $v \in \mathsf{var}(\Phi)$ and $C \in \mathsf{cla}(\Phi)$, there is an edge between v and C in G_Φ if and only if v or $\overline{v}$ belongs to the clause C. We say that Φ is a $K_{d,d}$-free formula, if G_Φ excludes $K_{d,d}$ as an induced subgraph [8].

Theorem 2. CC-Max-SAT *in* $K_{d,d}$-free *formulae admits a kernel of size* $\mathcal{O}(d4^{d^2}t^{d+1})$.

One may ask why not design a polynomial kernel for CC-Max-SAT in $K_{d,d}$-free formulae with parameter k or $k+d$. We would like to point out that CC-Max-SAT is W[1]-hard when parameterized by $k+d$, as explained in the below reduction (also the problem generalizes the Max Coverage problem). A simple reduction from a W[1]-hard problem, named Partial Vertex Cover is as follows [6]. Here, we want to cover the maximum number of edges in the input graph using k vertices. The construction of a formula for CC-Max-SAT is as follows. We will have a variable x_v for each vertex v in the input graph G. For an edge $e = \{u, v\}$, we will construct a clause $C_e = (x_v \vee x_u)$. The resulting formula will be $K_{2,2}$-free for a simple graph G. This implies that the CC-Max-SAT problem is W[1]-hard when parameterized by $k+d$.

2 Preliminaries

For a graph G, we denote its vertex set by $V(G)$ and its edge set by $E(G)$. We define a bipartite graph $G = (A, B, E)$ as a graph where $V(G)$ can be partitioned into two parts A and B, where each part is an independent set. For two sets A and B, $A \backslash B$ denotes the set difference of A and B, i.e., the set of elements in A, but not in B.

Let Φ be a CNF-formula. We use $\mathsf{var}(\Phi)$ and $\mathsf{cla}(\Phi)$ to denote the set of variables and clauses in Φ, respectively. We use G_Φ to denote the clause-variable incident bipartite graph of Φ. That is, the vertex set of G_Φ is $\mathsf{var}(\Phi) \uplus \mathsf{cla}(\Phi)$. For each $v \in \mathsf{var}(\Phi)$ and $C \in \mathsf{cla}(\Phi)$, there is an edge between v and C in G_Φ if and only if v or $\overline{v}$ belongs to the clause C.

An *all zero* assignment $\beta_\emptyset$ assigns false value to every variable of the input formula. With a slight abuse of terminology, for a clause $C \in \mathsf{cla}(\Phi)$, by $\mathsf{var}(C)$, we denote variables contained in C. For a subset Y of variables, $N^+_\Phi(Y)$ is the set of clauses in Φ that contains at least one variable from Y positively. We denote $N^-_\Phi(Y)$ as the set of clauses in Φ that contains at least one variable from Y negatively. For a subset Y of variables $N_\Phi(Y)$ is the set of clauses in Φ that contains at least one variable from Y either positively or negatively, that is the set $N^+_\Phi(Y) \cup N^-_\Phi(Y)$. For a clause C and variable $u \in \mathsf{var}(\Phi)$, by $C \setminus \{u\}$, we denote the clause C after removing the literal corresponding to u from C. For example let $C = u \vee v \vee w$, then $C \setminus \{u\} = v \vee w$. If $u \notin \mathsf{var}(C)$, then $C \setminus \{u\}$ is simply the clause C.

$K_{d,d}$ denotes the complete bipartite graph with each bipartition containing d vertices. A bipartite graph $G = (A, B, E)$ is $K_{d,d}$-free when G excludes $K_{d,d}$ as a subgraph (no set of d vertices in A together have d common neighbors in B).

3 FPT algorithm for CC-Max-SAT parameterized by t

In this section, we design an FPT algorithm for CC-Max-SAT problem parameterized by t, the number of satisfied clauses.

We use the algorithm for Partial Hitting Set given in the following proposition.

Proposition 1 (Bläser [2]). *There is an algorithm that given a universe U, a family of sets $\mathcal{F}$ over U, and positive integers k and t, runs in time $2^{\mathcal{O}(t)}n^{\mathcal{O}(1)}$, and checks whether there exists a k-sized subset of U, that hits at least t sets from $\mathcal{F}$. If such a subset exists, then the algorithm will output such a subset.*

We use the above proposition to get the following lemma.

Lemma 1. *There exists a randomized algorithm that solves the CC-Max-SAT problem and outputs a satisying assignment in time $2^{\mathcal{O}(t)}n^{\mathcal{O}(1)}$.*

Proof. We apply the procedure in Algorithm 1, 2^t times and if in any of the iteration it returns an assignment γ, then we output γ, otherwise we return NO. Next we prove the correctness of the algorithm.

Algorithm 1 Procedure: FPT Algorithm for CC-MAX-SAT

1: Randomly assign values to all the variables in $\mathsf{var}(\phi)$ from $\{0,1\}$. Let T be the set of variables assigned 1 and F be the set of variables assigned 0.
2: Delete all clauses C in $\mathsf{cla}(\phi)$ that contains a negative literal of a variable $v \in F$. Let t_1 be the number of deleted clauses and ϕ' be the resultant formula.
3: Construct a family $\mathcal{F}$ as follows.
4: **for** each clause $C \in \mathsf{cla}(\phi')$, **do**
5:     Add $\mathsf{var}(C) \setminus F$ to $\mathcal{F}$.
6: **end for**
7: Run the algorithm from Proposition 1 by converting to hitting set instance $(\mathcal{U}, \mathcal{F}, k, t - t_1)$ where $\mathcal{U} = \mathsf{var}(\phi') \setminus F$.
8: **if** the algorithm returns a NO. **then return** NO
9: **else** let S be the solution returned by the algorithm. We construct γ as follows.
10:     **if** $v \in S$ **then**
11:         let $\gamma(v) = 1$.
12:     **else** $\gamma(v) = 0$.
13:     **end if**
14:     **return** γ.
15: **end if**

Let (ϕ, k, t) be a YES instance for the problem. Let $C_1, C_2, \cdots C_t$ be t clauses that are satisfied by a particular assignment, say α. For each clause C_i, let x_i be the variable that is "responsible" for satisfying it by a feasible assignment α, that is assignment where at most k variables are assigned 1. By "responsible" we mean that the clause C_i is satisfied even if we give any other assignment to all the variables except x_i. Thus, $x_1, x_2, \cdots x_t$ are the "responsible" variables for the clauses $C_1, C_2, \cdots C_t$ by the assignment α. Let α_t be the assignment α restricted to these t variables. Now consider any random assignment $\alpha' : \mathsf{var}(\phi) \rightarrow \{0,1\}$ of $\mathsf{var}(\phi)$. Then the assignments α and α' agree on the variables $x_1, x_2, \cdots x_t$ with probability $\frac{1}{2^t}$. Let α' be the assignment obtained in step 1 of our procedure. Therefore with probaility $\frac{1}{2^t}$ the assignment α' satisfies the clauses $C_1, C_2, \cdots C_t$.

W.l.o.g. let $C_1, C_2, \cdots C_{t_1}$ be the set of clauses deleted in step 2 of our algorithm. Every clause C_i that is not deleted in step 2 must contain one of the variables from $x_1, x_2, \cdots x_t$ which is assigned 1. There are at most k such variables and hence the corresponding PARTIAL HITTING SET instance in step 7 will return a solution. Since we repeat the algorithm 2^t times, we get a success probability of $1 - \frac{1}{e}$.

Running Time: Observe that the running time of Procedure FPT Algorithm for CC-MAX-SAT depends on the algorithm used in step 7 which runs in time $2^{\mathcal{O}(t)} n^{\mathcal{O}(1)}$. As we repeat the Procedure 2^t times, our algorithm runs in time $2^{\mathcal{O}(t)} n^{\mathcal{O}(1)}$. □

We now derandomize the algorithm using Universal Sets. We deterministically construct a family $\mathcal{F}$ of functions $f : [n] \rightarrow [2]$ instead of randomly assigning values such that it is assured that one of the assignments when restricted to the

t variables $x_1, x_2, \cdots x_t$ matches with the assignment α_t. For this we state the following definitions.

Definition 1 ((n, ℓ)-universal set). *An (n, ℓ)-universal set is a family $\mathcal{U}$ of subsets of $[n]$ such that for every subset $S \subseteq [n]$ of size at most ℓ, the family $\{U \cap S : U \in \mathcal{U}\}$ contains all $2^{|S|}$ subsets of S.*

Proposition 2 (Naor et al. [11]). *There is an algorithm that given integers $n, \ell \in \mathbb{N}$, runs in time $2^\ell \ell^{\mathcal{O}(\log \ell)} n \log n$, and outputs an (n, ℓ)-universal set of cardinality at most $2^\ell \ell^{\mathcal{O}(\log \ell)} \log n$.*

We construct an (n, t)-universal set $\mathcal{U}$ and then for every element of $\mathcal{U}$ we construct an equivalent assignment by assigning $true$ to the variables represented by the elements in the subset and $false$ to the elements outside. By the definition of universal sets we will have an assignment which when restricted to the t responsible variables will be equal to α_t which corresponds to the good event in our random experiment. Thus, we have the following theorem.

Theorem 1. *There exists a deterministic algorithm that solves the* CC-Max-SAT *problem in time $2^{\mathcal{O}(t)} n \log n$.*

4 Polynomial Kernel for CC-Max-SAT in $K_{d,d}$-free formulas

In this section, we design a polynomial time kernelization algorithm for CC-Max-SAT where the input is a $K_{d,d}$-free formula, parameterized by the number of clauses to be satisfied by a solution – that is, the parameter is t. We begin by defining and recalling some notations.

An *all zero* assignment $\beta_\emptyset$ assigns false value to every variable of the input formula. With a slight abuse of terminology, for a clause $C \in \mathsf{cla}(\Phi)$, by $\mathsf{var}(C)$, we denote variables appearing in C. Recall that for a subset Y of variables $N^+_\Phi(Y)$ is the set of clauses in Φ that contains at least one variable from Y positively. We denote $N^-_\Phi(Y)$ as the set of clauses in Φ that contains at least one variable from Y negatively. We denote $d^+_\Phi(Y) = |N^+_\Phi(Y)|$ and $d^-_\Phi(Y) = |N^-_\Phi(Y)|$. For a subset Y of variables $N_\Phi(Y)$ is the set of clauses in Φ that contains at least one variable from Y either positively or negatively, that is the set $N^+_\Phi(Y) \cup N^-_\Phi(Y)$. For a clause C and variable $u \in \mathsf{var}(\Phi)$, by $C \setminus \{u\}$, we denote the clause C after removing the literal corresponding to u from C. For example let $C = u \vee v \vee w$, then $C \setminus \{u\} = v \vee w$. If $u \notin \mathsf{var}(C)$, then $C \setminus \{u\}$ is simply the clause C. Next, we give an outline of our kernel.

Outline of the kernel: Consider an instance (Φ, k, t) of CC-Max-SAT, where Φ is a $K_{d,d}$-free formula. Our kernelization algorithm works in three phases. In the first phase, we apply some simple sanity check reduction rules to eliminate trivial YES/NO instances of CC-Max-SAT. Reduction rules in this phase (1) upper bounds the frequency of variables in Φ by t, and (2) leads to an observation that any minimum weight assignment can satisfy at most $2t$ clauses. The above facts are useful to establish proofs in the next two phases.

Suppose (Φ, k, t) is a YES instance and let β be its minimum weight assignment and let $\mathcal{C}_\beta$ be the set of clauses satisfied by β. In the second phase, we bound the size of clauses in Φ by a function of t and d, say $f(t,d)$. Here, we use $K_{d,d}$-free property crucially. By the definition of $K_{d,d}$-free formula, a set of d variables cannot appear simultaneously in a set of d clauses. We generalize this idea together with the frequency bound on variables (obtained in phase one) to bound the size of a set of variables that appear together in $p \in [d-1]$ clauses by identifying and deleting some "redundant" variables.

For a set $Y \subseteq \mathsf{var}(\Phi)$, let $\mathsf{claInt}(Y)$ denote the set of all the clauses that contains all the variables in Y, that is the set $\{C \mid C \in \mathsf{cla}(\Phi), Y \subseteq \mathsf{var}(C)\}$.

For an intuition, suppose we have already managed to bound the size of every subset of $\mathsf{var}(\Phi)$, that appear together in at least two clauses, by $f(t,d)$. Now consider a set $Y \subseteq \mathsf{var}(\Phi)$ such that variables in Y appear together in at least one clause say C, that is $|\mathsf{claInt}(Y)| \geq 1$ and $C \in \mathsf{claInt}(Y)$. Now suppose that there is a clause $C^* \in \mathcal{C}_\beta \setminus \mathsf{claInt}(Y)$ such that $\mathsf{var}(C^*) \cap Y \neq \emptyset$. Then observe that for the variable set $\mathsf{var}(C^*) \cap Y$, C and C^* are common clauses. Now by considering the bound on sizes of sets of variables that have at least two common clauses (clauses in which they appear simultaneously), we obtain that $|\mathsf{var}(C^*) \cap Y|$ is also bounded by $f(t,d)$. We will now mark variables of all the clauses in $\mathcal{C}_\beta \setminus \mathsf{claInt}(Y)$ in Y and will conclude that if Y is sufficiently "large", then there exists a *redundant* variable in Y, that can be removed from Φ. Employing the above discussion, by repeatedly applying a careful deletion procedure, we manage to bound the size of sets of variables that have at least one common clause. By repeating these arguments we can show that to bound the size of sets of variables with at least 2 common clauses, all we require is to bound size of sets of variables with at least 3 common clauses. Thus, inductively, we bound size of sets of variables with $d-1$ common clauses, by using the fact that the input formula is $K_{d,d}$-free. Thus the algorithm starts for $d-1$ and applies a reduction rule to bound size of sets of variables with $d-1$ common clauses. Once we apply reduction rule for $d-1$ exhaustively, we apply for $d-2$ and by inductive application we reach the one common clause case. To apply reduction rule for $p \in [d-1]$ we assume that reduction rules for $d-1$ common clauses case have already been applied exhaustively. The bound on size of the sets of variables with one common clause also gives bound on the size of clause by $f(k,d)$.

Finally in the third phase, the algorithm applies a reduction rule to remove all the variables which are not among first $g(t,d)$ high positive degree variables and not among first $g(t,d)$ high negative degree variables when sorted in non decreasing ordering of their positive (negative) degrees, which together with the upper bound obtained on size frequency of variables and size of clauses obtained in phase one gives the desired kernel size.

We now formally introduce our reduction rules. Our algorithm applies each reduction rule exhaustively in the order in which they are stated. We begin by stating some simple sanity check reduction rules.

Reduction Rule 1. *If $k < 0$, or $k = 0$, $t \geq 1$ and $\beta_\emptyset$ satisfies less than t clauses in $\mathsf{cla}(\Phi)$, then return that (Φ, k, t) is a* NO *instance of CC-MAX-SAT.*

The safeness of Reduction Rule 1 follows from the fact that the cardinality of number of variables assigned 1 cannot be negative and since $k = 0$, all the variables must be assigned 0 values and an all zero assignment must satisfy at least t clauses for (Φ, k, t) to be a YES instance of CC-MAX-SAT.

Reduction Rule 2. *If at least one of the following holds, then return that (Φ, k, t) is a* YES *instance of CC-MAX-SAT.*

1. *$\beta_\emptyset$ satisfies at least t clauses in $\mathsf{cla}(\Phi)$.*
2. *$k \geq 0$ and there exists a variable $v \in \mathsf{var}(\Phi)$ such that $d^+_\Phi(v) \geq t$.*

The safeness of Reduction Rule 2 follows from the following facts: For the first condition, if an all zero assignment satisfies at least t clauses, then trivially (Φ, k, t) is a YES instance of CC-MAX-SAT. For the second condition, there exists an assignment which assigns v to true in case $d^+_\Phi(v) \geq t$, then the variable v alone can satisfy at least t clauses in $\mathsf{cla}(\Phi)$.

When none of the Reduction Rules 1 and 2 are applicable, we obtain the following observation.

Observation 1 *Consider a minimum weight assignment β of $\mathsf{var}(\Phi)$ that satisfies at least t clauses in $\mathsf{cla}(\Phi)$. Then the number of clauses satisfied by β is at most $2t$.*

Proof. As Reduction Rule 2 is no longer applicable β is not an all zero assignment. Let u be a variable that has been assigned true value by β. Now consider another assignment β', where for every $u' \in \mathsf{var}(\Phi), u' \neq u$, $\beta'(u') = \beta(u')$, and $\beta'(u)$ is false. As Reduction Rule 2 is no longer applicable, β' is also not an all zero assignment. Also, β' satisfies strictly less than t clauses, as otherwise it contradicts that β is a minimum weight assignment. Moreover, as Reduction Rule 2 is no longer applicable, $d^+_\Phi(u) < t$ and $d^-_\Phi(u) < t$. Notice that the difference between clauses satisfied by β and β' is exactly the clauses where u appears. The assignment β satisfies clauses where u appears positively, while β' satisfies clauses where u appears negatively. Above observations implies that β satisfies at most $t - d^-_\Phi(u) + d^+_\Phi(u) \leq 2t$ clauses. $\square$

Lemma 2. *Consider a set $Y \subseteq \mathsf{var}(\Phi)$. Let $|\mathsf{claInt}(Y)| = \ell$. If $|Y| \geq 2^\ell \cdot \tau + 1$, for some positive integer τ, then in polynomial time we can find sets $\widehat{Y}_{\mathsf{pos}}, \widehat{Y}_{\mathsf{neg}} \subseteq Y$ and a variable $\widehat{v} \in Y \setminus (\widehat{Y}_{\mathsf{pos}} \cup \widehat{Y}_{\mathsf{neg}})$ such that the following holds:*

1. *$|\widehat{Y}_{\mathsf{pos}}| = |\widehat{Y}_{\mathsf{neg}}| = \frac{\tau}{2}$.*

2. *Let* $\widehat{Y} = \widehat{Y}_{\mathsf{pos}} \cup \widehat{Y}_{\mathsf{neg}} \cup \{\widehat{v}\}$. *For every pair of variables* $u, u' \in \widehat{Y}$ *and every clause* $C \in \mathsf{claInt}(Y)$, u *appears in* C *positively (negatively) if and only if* u' *appears in* C *positively (negatively).*
3. *For every variable* $u \in \widehat{Y}_{\mathsf{pos}}$, $d^+_\Phi(\widehat{v}) \le d^+_\Phi(u)$.
4. *For every variable* $u \in \widehat{Y}_{\mathsf{neg}}$, $d^-_\Phi(\widehat{v}) \le d^-_\Phi(u)$.

Proof. Let $\mathsf{claInt}(Y) = \{C_1, \cdots, C_\ell\}$. For each $u \in Y$, we define a string Γ_u on $\{0,1\}$ of length ℓ by setting i-th bit of Γ_u as 1 (0) if u appears as positively (negatively) in C_i. By simple combinatorial arguments, we have that the number of different Γ strings that we can obtain are at most 2^ℓ. Since $|Y| \ge 2^\ell \cdot \tau + 1$, by pigeonhole principle there exists a set $Y' \subseteq Y$ of size at least $\tau + 1$ such that for every pair u, u' of variables in Y', $\Gamma_u = \Gamma_{u'}$. Also for every clause $C \in \mathsf{claInt}(Y)$, u appears in C positively (negatively) if and only if u' appears in C positively (negatively). Now we obtain $\widehat{Y}_{\mathsf{pos}}$ and $\widehat{Y}_{\mathsf{neg}}$ from Y'.

We let $\widehat{Y}_{\mathsf{pos}}$ be a subset of Y' of size $\frac{\tau}{2}$ such that for every $u \in \widehat{Y}_{\mathsf{pos}}$ and every $u' \in Y' \setminus \widehat{Y}_{\mathsf{pos}}$, $d^+_\Phi(u') \le d^+_\Phi(u)$. We let $\widehat{Y}_{\mathsf{neg}}$ be a subset of Y' of size $\frac{\tau}{2}$ such that for every $u \in \widehat{Y}_{\mathsf{neg}}$ and every $u' \in Y' \setminus \widehat{Y}_{\mathsf{neg}}$, $d^-_\Phi(u') \le d^-_\Phi(u)$, that is, $\widehat{Y}_{\mathsf{pos}}$ is the set of first $\frac{\tau}{2}$ variables in Y' when sorted by their positive degrees. Similarly $\widehat{Y}_{\mathsf{neg}}$ is the set of first $\frac{\tau}{2}$ variables in Y' when sorted by their negative degrees.

Observe that since $|\widehat{Y}_{\mathsf{pos}}| + |\widehat{Y}_{\mathsf{neg}}| < |Y'|$, therefore $Y' \setminus (\widehat{Y}_{\mathsf{pos}} \cup \widehat{Y}_{\mathsf{neg}}) \neq \emptyset$. We let $\widehat{v}$ be an arbitrary variable in $Y' \setminus (\widehat{Y}_{\mathsf{pos}} \cup \widehat{Y}_{\mathsf{neg}})$. By the above description, clearly $\widehat{v}, \widehat{Y}_{\mathsf{pos}}$, and $\widehat{Y}_{\mathsf{neg}}$ satisfies the properties stated in lemma and are computed in polynomial time. ☐

Next, we will describe reduction rules that help us bound the size of clauses in $\mathsf{cla}(\Phi)$. For each $p \in [d-1]$, we introduce Reduction Rule 3.p. We apply Reduction Rule 3.p in the increasing order of p. That is, first apply Reduction Rule 3.1 exhaustively, and for each $p \in [d-1] \setminus \{1\}$, apply Reduction Rule 3.p only if Reduction Rule 3.$(p-1)$ has been applied exhaustively. We apply our reduction rule on a "large" subset of $\mathsf{var}(\Phi)$. To quantify "large" we introduce the following definition.

Definition 2. *For each* $p \in [d-1]$, *we define an integer* z_p *as follows:*

- *If* $p = 1$, *then* $z_1 = 2^{d+1} \cdot (t(d-1)+1)$, *and*
- $z_p = 2^{d-p+2} \cdot (t \cdot z_{p-1} + 1)$, *otherwise.*

The following observation will be helpful to bound the size of our kernel.

Observation 2 $z_{d-1} \le 4^{d^2}(d \cdot t^d + 1)$.

Reduction Rule 3. *For each* $p \in [d-1]$ *we introduce* Reduction Rule 3.p *as follows. If there exists* $Y \subseteq \mathsf{var}(\Phi)$ *such that* $|\mathsf{claInt}(Y)| = d - p$ *and* $|Y| = z_p + 1$. *Use Lemma 2 to find sets* $\widehat{Y}_{\mathsf{pos}}, \widehat{Y}_{\mathsf{neg}} \subseteq Y$ *and a variable* $\widehat{v} \in Y \setminus (\widehat{Y}_{\mathsf{pos}} \cup \widehat{Y}_{\mathsf{neg}})$ *which satisfies properties stated in Lemma 2. Remove* $\widehat{v}$ *from* $\mathsf{var}(\Phi)$ *and return the instance* (Φ', k, t). *Here,* Φ' *is the formula with variable set* $\mathsf{var}(\Phi) \setminus \{\widehat{v}\}$ *and clause set* $\bigcup_{C \in \mathsf{cla}(\Phi)} C \setminus \{\widehat{v}\}$.

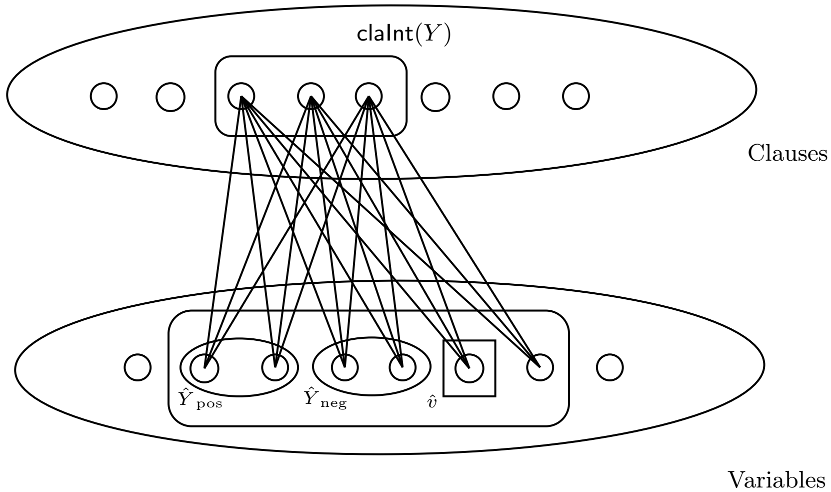

Fig. 2. A visual representation of Reduction Rule 3

Lemma 3. *Reduction Rule 3 is safe.*

Proof. To show that the lemma holds, we will show that (Φ, k, t) is a YES instance of CC-MAX-SAT if and only if (Φ', k, t) is a YES instance of CC-MAX-SAT, for each $p \in [d-1]$. We prove the lemma by induction on p.
Base Case: $p = 1$. We have $z_1 = 2^{d+1} \cdot (t(d-1)+1)$. By Lemma 2 we have $|\widehat{Y}_{\mathsf{pos}}| = |\widehat{Y}_{\mathsf{neg}}| = 2t(d-1)+2$. In the forward direction suppose that (Φ, k, t) is a YES instance of CC-MAX-SAT and let β be its minimum weight assignment. Let $\mathcal{C}_\beta \subseteq \mathsf{cla}(\Phi)$ be the set of clauses satisfied by β and $\widetilde{\mathcal{C}_\beta} = \mathcal{C}_\beta \setminus \mathsf{claInt}(Y)$ be the set of clauses satisfied by β but not in $\mathsf{claInt}(Y)$. Observe the following:

1. By Observation 1, the number of clauses satisfied by β is at most $2t$, that is $|\mathcal{C}_\beta| \leq 2t$.
2. Suppose that $C \in \widetilde{\mathcal{C}_\beta}$. Then, since Φ is a $K_{d,d}$-free formula and $|\mathsf{claInt}(Y) \cup \{C\}| = d$, we have $|\mathsf{var}(C) \cap Y| \leq d-1$. As otherwise the set of variables $\mathsf{var}(C) \cap Y$ and set of clauses $\mathsf{claInt}(Y) \cup \{C\}$ will contradict $K_{d,d}$-free property.

By (1) and (2) the set of variables in Y, that appear in clauses in the set $\widetilde{\mathcal{C}_\beta}$ is bounded by $2t(d-1)$. Hence the set of variables in $\widehat{Y}_{\mathsf{pos}}$ and in $\widehat{Y}_{\mathsf{neg}}$, that appear in clauses in the set $\widetilde{\mathcal{C}_\beta}$ is bounded by $2t(d-1)$. That is $|\bigcup_{C \in \widetilde{\mathcal{C}_\beta}} \mathsf{var}(C) \cap \widehat{Y}_{\mathsf{pos}}| \leq 2t(d-1)$ and $|\bigcup_{C \in \widetilde{\mathcal{C}_\beta}} \mathsf{var}(C) \cap \widehat{Y}_{\mathsf{neg}}| \leq 2t(d-1)$.

Recall that $|\widehat{Y}_{\mathsf{pos}}| = |\widehat{Y}_{\mathsf{neg}}| = 2t(d-1)+2$. Therefore, there exists two variables say $w_1, w_2 \in \widehat{Y}_{\mathsf{pos}}$ such that w_1, w_2 do not appear in any clause in $\widetilde{\mathcal{C}_\beta}$. Similarly, there exists two variables say $u_1, u_2 \in \widehat{Y}_{\mathsf{neg}}$ such that u_1, u_2 do not appear in any clause in $\widetilde{\mathcal{C}_\beta}$. That is every clause that is satisfied by β and contains any of w_1, w_2, u_1, u_2 is contained in $\mathsf{claInt}(Y)$. This also implies that every clause that is satisfied by any of the variables w_1, w_2, u_1, u_2 in assignment β, is contained in $\mathsf{claInt}(Y)$. Now consider the following cases:

Case 1: $\beta(\widehat{v}) = 1$. First we claim that w_1 and w_2 are both set to false by β. Suppose for a contradiction that one of them say w_1 is set to true, that is $\beta(w_1) = 1$. By the properties of $\widehat{Y}_{\mathsf{pos}}$, and $\widehat{v}$ (See Lemma 2), in every clause $C \in \mathsf{claInt}(Y)$, either both $\widehat{v}, w_1$ appear positively in C or both $\widehat{v}, w_1$ appear negatively in C. Further, since w_1 can only satisfy clauses in $\mathsf{claInt}(Y)$ by assignment β, we have that w_1 satisfies same set of clauses as $\widehat{v}$ in $\mathsf{claInt}(Y)$. That is $N_\Phi(w_1) \cap \mathcal{C}_\beta = N_\Phi(\widehat{v}) \cap \mathcal{C}_\beta$. In this case, we can obtain another assignment of smaller weight by setting w_1 to false, which contradicts that β is a minimum weight assignment. By similar arguments we can show that w_2 is set to false by β.

Now we will construct an assignment β' for variables $\mathsf{var}(\Phi') = \mathsf{var}(\Phi) \setminus \{\widehat{v}\}$, by setting the value of w_1 to true and we will show that β' satisfies as many clauses in Φ' as satisfied by β in Φ. We define β' formally as follows: $\beta'(w_1) = 1$, and for every $u \in \mathsf{var}(\Phi) \setminus \{\widehat{v}, w_1\}$, $\beta'(u) = \beta(u)$. Notice that $\mathcal{C}_\beta$ comprises of the following set of clauses satisfied by β (i) clauses that do not contain variables $\widehat{v}, w_1$, (ii) clauses that are in $\mathsf{claInt}(Y)$ and that contains variable $\widehat{v}$ positively, (iii) clauses that are in $\mathsf{claInt}(Y)$ and that contains variable w_1 negatively, and lastly (iv) clauses that are not in $\mathsf{claInt}(Y)$ and that contains variable $\widehat{v}$ positively.

Clearly $\mathsf{cla}(\Phi')$ contains every clause in the set (i) and they are also satisfied by β'. For every clause $C \in \mathsf{cla}(\Phi)$ that contain $\widehat{v}$, we have a clause $C \setminus \{\widehat{v}\}$ in $\mathsf{cla}(\Phi')$. If C is a clause in the set (ii), then C contains $\widehat{v}$ positively and thus also contains w_1 positively. Therefore, $C \setminus \{\widehat{v}\}$ is satisfied by β'. Hence for every clause in the set (ii), we have a clause in $\mathsf{cla}(\Phi')$ satisfied by w_1 in β'. Next, consider a clause C in the set (iii), then C contains w_1 negatively and thus also contains w_2 negatively. As argued before $\beta(w_2) = \beta'(w_2) = 0$. Therefore, $C \setminus \{\widehat{v}\}$ is satisfied by w_2 in β'. Hence for every clause in set (iii), we have a clause in $\mathsf{cla}(\Phi')$ satisfied by β'.

Now we are only remaining to show that β' compensate for the clauses in the set (iv) for Φ'. For that purpose recall that we have $d^+_\Phi(\widehat{v}) \le d^+_\Phi(w_1)$, by the properties of $\widehat{Y}_{\mathsf{pos}}$, and $\widehat{v}$ (See Lemma 2). Therefore $|N^+_\Phi(w_1)| \ge |N^+_\Phi(\widehat{v})|$, and hence $|N^+_{\Phi'}(w_1)| \ge |N^+_\Phi(\widehat{v})|$. Also $(N^+_\Phi(w_1) \cap \mathcal{C}_\beta) \setminus \mathsf{claInt}(Y) = \emptyset$. That is, the clauses which contains w_1 positively and are not in $\mathsf{claInt}(Y)$ were not satisfied by β. As β' sets w_1 to true, now clauses in $N^+_{\Phi'}(w_1)$ are satisfied by β'. We obtain the following:

$$|(N^+_\Phi(w_1) \cap \mathcal{C}_\beta)| - |N^+_\Phi(w_1) \cap \mathsf{claInt}(Y)| \ge |(N^+_\Phi(\widehat{v}) \cap \mathcal{C}_\beta)| - |N^+_\Phi(\widehat{v}) \cap \mathsf{claInt}(Y)|.$$

$$|(N^+_{\Phi'}(w_1) \cap \mathcal{C}_{\beta'})| - |N^+_\Phi(w_1) \cap \mathsf{claInt}(Y)| \ge |(N^+_\Phi(\widehat{v}) \cap \mathcal{C}_\beta)| - |N^+_\Phi(\widehat{v}) \cap \mathsf{claInt}(Y)|.$$

All the above discussion concludes that the number of clauses satisfied by β' in Φ' are at least the number of clauses satisfied by β in Φ. Hence, (Φ', k, t) is a YES instance of CC-Max-SAT.

Case 2: $\beta(\widehat{v}) = 0$. We can show that at least one of u_1, u_2 is set to false by β, by using analogous arguments that were used in Case 1 to show that when

$\widehat{v}$ is set to true then both w_1 and w_2 are set to false. Without loss of generality suppose that u_1 is set to false by β. We will show that assignment β restricted to $\mathsf{var}(\Phi') = \mathsf{var}(\Phi) \setminus \{\widehat{v}\}$ satisfies as many clauses in Φ' as satisfied by β in Φ. We define β' as β restricted to $\mathsf{var}(\Phi')$ formally as follows: for every $u \in \mathsf{var}(\Phi) \setminus \{\widehat{v}\}$, $\beta'(u) = \beta(u)$. Notice that $\mathcal{C}_\beta$ comprises of the following set of clauses satisfied by β (i) clauses that do contain variable $\widehat{v}$, (ii) clauses that are in $\mathsf{claInt}(Y)$ and that contains variable $\widehat{v}$ negatively, and lastly (iii) clauses that are not in $\mathsf{claInt}(Y)$ and that contains variable $\widehat{v}$ negatively.

Clearly $\mathsf{cla}(\Phi')$ contains every clause in the set (i) and they are also satisfied by β'. For every clause $C \in \mathsf{cla}(\Phi)$ that contain $\widehat{v}$, we have a clause $C \setminus \{\widehat{v}\}$ in $\mathsf{cla}(\Phi')$. If C is a clause in the set (ii), then C contains $\widehat{v}$ negatively and thus also contains u_1 negatively. Therefore, $C \setminus \{\widehat{v}\}$ is satisfied by β'. Hence for every clause in the set (ii), we have a clause in $\mathsf{cla}(\Phi')$ satisfied by u_1 in β'. Now we are only remaining to consider the clauses in the set (iii). That is the set of clauses that are not in $\mathsf{claInt}(Y)$ and contains $\widehat{v}$ negatively. We claim that there is no clause in set (iii), that is $N^-_\Phi(\widehat{v}) \setminus \mathsf{claInt}(Y) = \emptyset$.

To prove the claim first recall that we have $d^-_\Phi(\widehat{v}) \leq d^-_\Phi(u_1)$, by the properties of $\widehat{Y}_{\mathsf{neg}}$, and $\widehat{v}$ (See Lemma 2). Therefore $|N^-_\Phi(u_1)| \geq |N^-_\Phi(v)|$. Also $N^-_\Phi(u_1) \cap \mathsf{claInt}(Y) = N^-_\Phi(\widehat{v}) \cap \mathsf{claInt}(Y)$. Further, $(N^-_\Phi(u_1) \cap \mathcal{C}_\beta) \setminus \mathsf{claInt}(Y) = \emptyset$. Therefore, $N^-_\Phi(u_1) \setminus \mathsf{claInt}(Y) = \emptyset$, as u_1 is set to false by β. If $(N^-_\Phi(\widehat{v}) \cap \mathcal{C}_\beta) \setminus \mathsf{claInt}(Y) \neq \emptyset$, then $N^-_\Phi(\widehat{v}) \setminus \mathsf{claInt}(Y) \neq \emptyset$. Therefore $|N^-_\Phi(u_1) \setminus \mathsf{claInt}(Y)| < |N^-_\Phi(\widehat{v}) \setminus \mathsf{claInt}(Y)|$. Hence, the following holds:

$$d^-_\Phi(u_1) = |(N^-_\Phi(u_1) \cap \mathsf{claInt}(Y))| + |(N^-_\Phi(u_1) \cap \mathsf{claInt}(Y))|,$$

$$d^-_\Phi(u_1) = |(N^-_\Phi(u_1) \cap \mathsf{claInt}(Y))| + |(N^-_\Phi(\widehat{v}) \cap \mathsf{claInt}(Y))|,$$

$$d^-_\Phi(u_1) < |(N^-_\Phi(\widehat{v}) \cap \mathsf{claInt}(Y))| + |(N^-_\Phi(\widehat{v}) \cap \mathsf{claInt}(Y))|.$$

Thus $d^-_\Phi(u_1) < d^-_\Phi(\widehat{v})$, a contradiction. All the above discussion concludes that the number of clauses satisfied by β' in Φ' are at least the number of clauses satisfied by β in Φ. Hence, (Φ', k, t) is a YES instance of CC-Max-SAT.

It is easy to see that in the backward direction if (Φ', k, t) is a YES instance of CC-Max-SAT then (Φ, k, t) is a YES instance of CC-Max-SAT. As any assignment β' of $\mathsf{var}(\Phi')$ can be extended to an assignment β of Φ by setting $\widehat{v}$ to false and assigning every variable $u \neq \widehat{v}$ as $\beta(u)$. By the definition of Φ', β satisfies as many clauses as β' and weight of β is equal to weight of β'.

Induction Hypothesis: Assume that Reduction Rule 3.p is safe for all $p < q$, $q \in [d-2]$.

Inductive Case: $p = q$. We have $z_p = 2^{d-p+2} \cdot (t \cdot z_{p-1} + 1) + 1$. By Lemma 2 we have $|\widehat{Y}_{\mathsf{pos}}| = |\widehat{Y}_{\mathsf{neg}}| = 2t \cdot z_{p-1} + 2$. In the forward direction suppose that (Φ, k, t) is a YES instance of CC-Max-SAT and let β be its minimum weight assignment. Let $\mathcal{C}_\beta \subseteq \mathsf{cla}(\Phi)$ be the set of clauses satisfied by β and $\widetilde{\mathcal{C}_\beta} = \mathcal{C}_\beta \setminus \mathsf{claInt}(Y)$ be the set of clauses satisfied by β but not in $\mathsf{claInt}(Y)$. Observe the following:

1. By Observation 1, the number of clauses satisfied by β is at most $2t$, that is $|\mathcal{C}_\beta| \leq 2t$.

2. Suppose that $C \in \widetilde{\mathcal{C}_\beta}$. Then, since Reduction Rule 3.$(p-1)$ is not applicable and $|\mathsf{claInt}(Y) \cup \{C\}| = d-p+1 = d-(p-1)$, we have $|\mathsf{var}(C) \cap Y| \leq z_{p-1}$. As otherwise the set of variables $\mathsf{var}(C) \cap Y$ and set of clause $\mathsf{claInt}(Y) \cup \{C\}$ will contradict that Reduction Rule 3.$(p-1)$ is not applicable.

By (1) and (2) the number of variables in Y, that appear in clauses in the set $\widetilde{\mathcal{C}_\beta}$ is bounded by $2t \cdot z_{p-1}$. Hence the number of variables in $\widehat{Y}_{\mathsf{pos}}$ and in $\widehat{Y}_{\mathsf{neg}}$, that appear in clauses in the set $\widetilde{\mathcal{C}_\beta}$ is bounded by $2t \cdot z_{p-1}$. That is $|\bigcup_{C \in \widetilde{\mathcal{C}_\beta}} \mathsf{var}(C) \cap \widehat{Y}_{\mathsf{pos}}| \leq 2t \cdot z_{p-1}$ and $|\bigcup_{C \in \widetilde{\mathcal{C}_\beta}} \mathsf{var}(C) \cap \widehat{Y}_{\mathsf{neg}}| \leq 2t \cdot z_{p-1}$.

Recall that $|\widehat{Y}_{\mathsf{pos}}| = |\widehat{Y}_{\mathsf{neg}}| = 2t \cdot z_{p-1} + 2$. Therefore, there exists two variables say $w_1, w_2 \in \widehat{Y}_{\mathsf{pos}}$ such that w_1, w_2 do not appear in any clause in $\widetilde{\mathcal{C}_\beta}$. Similarly, there exists two variables say $u_1, u_2 \in \widehat{Y}_{\mathsf{neg}}$ such that u_1, u_2 do not appear in any clause in $\widetilde{\mathcal{C}_\beta}$. Now by analogous arguments as in Case 1 and Case 2 of the base case, it follows that (Φ', k, t) is a YES instance of CC-Max-SAT. Backward direction also follows similar to the base case. This completes the proof. ☐

When Reduction Rule 3 is no longer applicable, we have that every set $Y \subseteq \mathsf{var}(\Phi)$ such that $|\mathsf{claInt}(Y)| \geq d-(d-1) = 1$, satisfies $|Y| \leq z_{d-1}$. In other words for every clause $C \in \mathsf{cla}(\Phi)$, $|\mathsf{var}(C)| \leq z_{d-1}$. We record this observation in the following.

Observation 3 *When Reduction Rules 1-3 are not applicable, then for every clause $C \in \mathsf{cla}(\Phi)$, $|\mathsf{var}(C)| \leq z_{d-1}$.*

For stating our next reduction rule, we define two sets $\widehat{V}_{\mathsf{pos}}, \widehat{V}_{\mathsf{neg}} \subseteq \mathsf{var}(\Phi)$ of size $\min\{n, z_{d-1}+1\}$ with the following properties: (1) For every variable $u \in \widehat{V}_{\mathsf{pos}}$ and every variable $u' \in \mathsf{var}(\Phi) \setminus \widehat{V}_{\mathsf{pos}}$, $d^+_\Phi(u) \geq d^+_\Phi(u')$. (2) For every variable $u \in \widehat{V}_{\mathsf{neg}}$ and every variable $u' \in \mathsf{var}(\Phi) \setminus \widehat{V}_{\mathsf{neg}}$, $d^-_\Phi(u) \geq d^-_\Phi(u')$. Clearly, if $\mathsf{var}(\Phi) \geq 2t \cdot z_{d-1} + 3$, then by pigeonhole principle $\mathsf{var}(\Phi) \setminus (\widehat{V}_{\mathsf{pos}} \cup \widehat{V}_{\mathsf{neg}}) \neq \emptyset$.

Reduction Rule 4. *If $|\mathsf{var}(\Phi)| \geq 2t \cdot z_{d-1} + 3$, then let $u \in \mathsf{var}(\Phi) \setminus (\widehat{V}_{\mathsf{pos}} \cup \widehat{V}_{\mathsf{neg}})$. Remove u from $\mathsf{var}(\Phi)$ and return the instance (Φ', k, t). Here Φ' is the formula with variable set $\mathsf{var}(\Phi) \setminus \{u\}$ and clause set $\bigcup_{C \in \mathsf{cla}(\Phi)} C \setminus \{u\}$.*

Lemma 4. *Reduction Rule 4 is safe.*

Proof. In the forward direction suppose that (Φ, k, t) is a YES instance of CC-Max-SAT. Let β be its minimum weight assignment and let X be the set of clauses satisfied by β. Since Reduction Rule 2 is not applicable β is not an all zero assignment.

(1) By Observation 1, $|X| \leq 2t$.

(2)By Observation 3, for every $C \in \mathsf{cla}(\Phi)$, $|\mathsf{var}(C)| \leq z_{d-1}$.

Let $Y = \bigcup_{C \in X} \mathsf{var}(C)$, then by (1) and (2), $|Y| \leq 2t \cdot z_{d-1}$. Therefore, there exists a variable say $w_1 \in \widehat{V}_{\mathsf{pos}}$ and there exists a variable say $w_2 \in \widehat{V}_{\mathsf{neg}}$ such that $w_1, w_2 \notin Y$ and hence, $N_\Phi(w_1) \cap X = N_\Phi(w_2) \cap X = \emptyset$. Since β is a minimum weight assignment, $\beta(w_1) = \beta(w_2) = 0$.

Case 1: $\beta(u) = 1$. We obtain another assignment by setting the value of w_1 to true. That is we construct a new assignment β' of $\mathsf{var}(\Phi') = \mathsf{var}(\Phi) \setminus \{u\}$ as follows: $\beta'(w_1) = 1$, and for every $v \in \mathsf{var}(\Phi) \setminus \{u, w_1\}$, $\beta'(v) = \beta(v)$. We have $d^+_\Phi(u) \leq d^+_\Phi(w_1)$ (by the definition of $\widehat{V}_{\mathsf{pos}}$). Let $X' = (X \setminus N^+_\Phi(u)) \cup N^+_\Phi(w_1)$ and $X'' = \bigcup_{C \in X'} C \setminus \{u\}$. Then observe that X'' is the set of clauses satisfied by β'. Also $|X''| = |X'| \geq |X| \geq t$. This implies that β' is a solution to (Φ', k, t) of CC-Max-SAT.

Case 2: $\beta(u) = 0$. We have $d^-_\Phi(u) \leq d^-_\Phi(w_2)$ (by the definition of $\widehat{V}_{\mathsf{neg}}$) and $N^-_\Phi(w_2) \cap X = \emptyset$. As both u, w_2 are set to false by β, $N^-_\Phi(w_2) \setminus X = N^-_\Phi(u) \setminus X = \emptyset$. Therefore $N^-_\Phi(u) \cap X = \emptyset$, as otherwise $|N^-_\Phi(u) \cap X| > |N^-_\Phi(w_2) \cap X|$ which contradicts $d^-_\Phi(u) \leq d^-_\Phi(w_2)$. Then observe that X'' is the set of clauses satisfied by β restricted to $\mathsf{var}(\Phi) \setminus \{u\} = \mathsf{var}(\Phi')$. Also $|X''| = |X'| = |X| \geq t$. This implies that assignment β restricted to $\mathsf{var}(\Phi')$ is a solution to (Φ', k, t) of CC-Max-SAT.

It is easy to see that in the backward direction if (Φ', k, t) is a YES instance of CC-Max-SAT then (Φ, k, t) is a YES instance of CC-Max-SAT. As any assignment β' of $\mathsf{var}(\Phi)$ can be extended to an assignment β of Φ by setting u to false and assigning every variable $v \neq u$ as $\beta(v)$. Observe that β satisfies as many clauses as β' and weight of β is equal to weight of β'. This completes the proof. □

Observe that Reduction Rules 1,2 and 4 can be applied in polynomial time. Reduction Rule 3 can be applied in $n^{\mathcal{O}(4^{d^2})}$ time. Each of our reduction rule is applicable only polynomial many times. Hence, the kernelization algorithm runs in polynomial time. Each of our reduction rule is safe. When Reduction Rules 1-4 are no longer applicable, the size of the set $\mathsf{var}(\Phi)$ is bounded by $2t(z_{d-1})+2$, and number of clauses a variable appear in is bounded by $2t$, and there are no variables which do not appear in any clause. Therefore by using Observation 1, the number of variables and clauses in Φ is bounded by $\mathcal{O}(d4^{d^2+1}t^{d+1})$.

Theorem 2. CC-Max-SAT *in* $K_{d,d}$-free *formulae admits a kernel of size* $\mathcal{O}(d4^{d^2}t^{d+1})$.

References

1. Agrawal, A., Choudhary, P., Jain, P., Kanesh, L., Sahlot, V., Saurabh, S.: Hitting and covering partially. In: COCOON. pp. 751–763 (2018)
2. Bläser, M.: Computing small partial coverings. Inf. Process. Lett. **85**(6), 327–331 (2003). https://doi.org/10.1016/S0020-0190(02)00434-9, https://doi.org/10.1016/S0020-0190(02)00434-9
3. Cygan, M., Fomin, F.V., Kowalik, L., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: Parameterized Algorithms. Springer (2015). https://doi.org/10.1007/978-3-319-21275-3, https://doi.org/10.1007/978-3-319-21275-3
4. Dom, M., Lokshtanov, D., Saurabh, S.: Kernelization lower bounds through colors and ids. ACM Trans. Algorithms **11**(2), 13:1–13:20 (2014)

5. Feige, U.: A threshold of ln n for approximating set cover. J. ACM **45**(4), 634–652 (1998). https://doi.org/10.1145/285055.285059, https://doi.org/10.1145/285055.285059
6. Guo, J., Niedermeier, R., Wernicke, S.: Parameterized complexity of vertex cover variants. Theory Comput. Syst. **41**(3), 501–520 (2007). https://doi.org/10.1007/s00224-007-1309-3, https://doi.org/10.1007/s00224-007-1309-3
7. Jain, P., Kanesh, L., Panolan, F., Saha, S., Sahu, A., Saurabh, S., Upasana, A.: Parameterized approximation scheme for biclique-free max k-weight SAT and max coverage. In: Bansal, N., Nagarajan, V. (eds.) Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023. pp. 3713–3733. SIAM (2023)
8. Lokshtanov, D., Panolan, F., Ramanujan, M.S.: Backdoor sets on nowhere dense SAT. In: Bojanczyk, M., Merelli, E., Woodruff, D.P. (eds.) 49th International Colloquium on Automata, Languages, and Programming, ICALP 2022, July 4-8, 2022, Paris, France. LIPIcs, vol. 229, pp. 91:1–91:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2022). https://doi.org/10.4230/LIPIcs.ICALP.2022.91, https://doi.org/10.4230/LIPIcs.ICALP.2022.91
9. Manurangsi, P.: Tight running time lower bounds for strong inapproximability of maximum k-coverage, unique set cover and related problems (via t-wise agreement testing theorem). In: Chawla, S. (ed.) Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020. pp. 62–81. SIAM (2020). https://doi.org/10.1137/1.9781611975994.5, https://doi.org/10.1137/1.9781611975994.5
10. Muise, C.J., Beck, J.C., McIlraith, S.A.: Optimal partial-order plan relaxation via maxsat. J. Artif. Intell. Res. **57**, 113–149 (2016). https://doi.org/10.1613/jair.5128, https://doi.org/10.1613/jair.5128
11. Naor, M., Schulman, L.J., Srinivasan, A.: Splitters and near-optimal derandomization. In: 36th Annual Symposium on Foundations of Computer Science, Milwaukee, Wisconsin, USA, 23-25 October 1995. pp. 182–191. IEEE Computer Society (1995). https://doi.org/10.1109/SFCS.1995.492475, https://doi.org/10.1109/SFCS.1995.492475
12. Skowron, P., Faliszewski, P.: Chamberlin-courant rule with approval ballots: Approximating the maxcover problem with bounded frequencies in FPT time. J. Artif. Intell. Res. **60**, 687–716 (2017). https://doi.org/10.1613/jair.5628, https://doi.org/10.1613/jair.5628
13. Sviridenko, M.: Best possible approximation algorithm for MAX SAT with cardinality constraint. Algorithmica **30**(3), 398–405 (2001)
14. Telle, J.A., Villanger, Y.: FPT algorithms for domination in sparse graphs and beyond. Theor. Comput. Sci. **770**, 62–68 (2019). https://doi.org/10.1016/j.tcs.2018.10.030, https://doi.org/10.1016/j.tcs.2018.10.030
15. Zhang, L., Bacchus, F.: MAXSAT heuristics for cost optimal planning. In: Hoffmann, J., Selman, B. (eds.) Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada. pp. 1846–1852. AAAI Press (2012). https://doi.org/10.1609/aaai.v26i1.8373, https://doi.org/10.1609/aaai.v26i1.8373