Theoretical Computer Science 506 (2013) 115–121

Contents lists available at SciVerse ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Solving MIN ONES 2-SAT as fast as VERTEX COVER ☆

Neeldhara Misra [a], N.S. Narayanaswamy [c,1], Venkatesh Raman [b,*], Bal Sri Shankar [b]

[a] Department of Computer Science and Automation, Indian Institute of Science, Bangalore, India
[b] Institute of Mathematical Sciences, Chennai, India
[c] Indian Institute of Technology, Madras, India

ARTICLE INFO

Article history:
Received 12 March 2012
Received in revised form 26 April 2013
Accepted 18 July 2013
Communicated by J. Kratochvil

Keywords:
Min ones 2-sat
Vertex cover
Linear programming relaxation
Parameterized complexity

ABSTRACT

The problem of finding a satisfying assignment that minimizes the number of variables that are set to 1 is NP-complete even for a satisfiable 2-SAT formula. We call this problem MIN ONES 2-SAT. It generalizes the well-studied problem of finding the smallest vertex cover of a graph, which can be modeled using a 2-SAT formula with no negative literals. The natural parameterized version of the problem asks for a satisfying assignment of weight at most k. In this paper, we present a polynomial-time reduction from MIN ONES 2-SAT to VERTEX COVER without increasing the parameter and ensuring that the number of vertices in the reduced instance is *equal* to the number of variables of the input formula. Consequently, we conclude that this problem also has a simple 2-approximation algorithm and a $2k - c\log k$-variable kernel subsuming (or, in the case of kernels, improving) the results known earlier. Further, the problem admits algorithms for the parameterized and optimization versions whose runtimes will always match the runtimes of the best-known algorithms for the corresponding versions of vertex cover.

Finally we show that the optimum value of the LP relaxation of the MIN ONES 2-SAT and that of the corresponding VERTEX COVER are the same. This implies that the (recent) results of VERTEX COVER version parameterized above the optimum value of the LP relaxation of VERTEX COVER carry over to the MIN ONES 2-SAT version parameterized above the optimum of the LP relaxation of MIN ONES 2-SAT.

1. Introduction and motivation

Satisfiability is a fundamental problem that encodes several computational problems. Variations of the problem appear as canonical complete problems for several complexity classes. While it is well known that the satisfiability of a formula in CNF form is a canonical NP-complete problem, testing whether a CNF formula has a satisfying assignment with weight[2] k is a canonical complete problem for the parameterized complexity class $W[2]$ [7]. If the number of variables in each clause is bounded, it is a canonical $W[1]$-complete problem [7]. These results imply that it is unlikely that these problems are *fixed-parameter tractable* (FPT). In other words, it is unlikely that they have an algorithm with running time $f(k)n^{O(1)}$ on input formulas of size n.

☆ A preliminary version of this work appeared in the Mathematical Foundations of Computer Science (MFCS) 2010, LNCS 6281, 2010, pp. 549–555.

* Corresponding author.
E-mail addresses: mail@neeldhara.com (N. Misra), swamy@cse.iitm.ac.in (N.S. Narayanaswamy), vraman@imsc.res.in (V. Raman), balsri@imsc.res.in (B.S. Shankar).

[1] Part of this work was done when the author was visiting The Institute of Mathematical Sciences, on sabbatical from IIT Madras.
[2] The *weight* of an assignment is the number of variables assigned 1 by the assignment.

0304-3975/$ – see front matter
http://dx.doi.org/10.1016/j.tcs.2013.07.019

On the other hand, if the question is whether a d-CNF formula (for fixed d) has a satisfying assignment with weight *at most* k, then this generalizes the well-studied d-hitting set problem and turns out to be fixed-parameter tractable with the weight as a parameter ([17,16], see also Section 2). When we restrict our attention to 2-CNF formulas (MIN ONES 2-SAT) this problem generalizes the well-studied VERTEX COVER problem. Given a graph $G = (V, E)$, introduce a variable v for every vertex $v \in V$ and consider the formula $\bigwedge(u \vee v)$, where the $\bigwedge$ runs over all pairs u, v of variables such that (u, v) is in E. Then a satisfying assignment of weight k corresponds to a vertex cover of size k and vice versa. However, notice that we do not require negated literals to encode VERTEX COVER using 2-CNF formulas, and thus it appears that MIN ONES 2-SAT is a more general version of the vertex cover problem.

1.1. Related work

Gusfield and Pitt [8] considered this MIN ONES 2-SAT problem and gave a 2-approximation algorithm. The algorithm follows a greedy approach and gives a solution whose weight is at most twice the optimum (assuming that the formula is satisfiable). As satisfiability of 2-CNF-SAT is well known to be polynomial time solvable, we can assume without loss of generality that the given 2-SAT formula is satisfiable. See [24] for efficient exact exponential algorithms for MIN ONES 2-SAT.

One approach to design a 2-approximation algorithm for the optimization version and a $2k$-variable kernel for the parameterized version of VERTEX COVER (see Section 2 for definitions of parameterized complexity) is through linear programming. The minimum vertex cover problem can be formulated as an integer linear programming by introducing a boolean (0–1) variable x_i for every vertex $i \in V$, and by writing the constraint $x_i + x_j \geqslant 1$ $\forall (i, j) \in E$. The objective function is to minimize $\sum_{i \in V} x_i$. The linear programming relaxation of the problem relaxes the variables x_i's to take values from the interval $[0, 1]$ (instead of just values 0 and 1). A classical theorem due to Nemhauser and Trotter [20] states the following (here $n = |V|$):

Theorem 1. *(See [20].)*

- *There exists an optimum solution $x^* = (x_1^*, x_2^*, \ldots, x_n^*)$ to the linear programming relaxation of the integer programming formulation of* VERTEX COVER *where the variables take the values* 0, 1/2 *or* 1, *and such a solution can be found in polynomial time.*
- *There exists an optimum solution to* VERTEX COVER (*i.e. to the integer programming formulation*) *that contains all vertices i such that $x_i^* = 1$ and none of the vertices i such that $x_i^* = 0$.*

By solving the linear program, and by including all vertices with $x_i^* = 1$ into the solution and by deleting all vertices with $x_i^* = 1$ or 0, one can obtain a 2-approximation algorithm, and a $2k$-vertex kernel for VERTEX COVER. Hochbaum et al. [9] showed that the classical Nemhauser–Trotter theorem for vertex cover [20] holds for MIN ONES 2-SAT as well. This implies a 2-approximation algorithm for the optimization version, and a $2k$-variable kernel for the parameterized version of MIN ONES 2-SAT as well.

There is a reduction from 2-SAT to VERTEX COVER, pointed out by Seffi Naor (see [10]). This reduction takes an instance F of MIN ONES 2-SAT on n variables, and first computes the *closure* of the clauses (see Section 3). The closure consists of all the original clauses and clauses $(x \vee y)$ whenever $(x \vee z)$ and $(y \vee \bar{z})$ appear in the original clauses of F. From the closure of F, a graph $G(F)$ is constructed that has one vertex for every literal participating in F and an edge between a pair of literals whenever they appear together in a clause of the closure of F, and an edge $(x, \bar{x})$ for every variable x. It is shown that any satisfying assignment for F corresponds to a vertex cover of size n in $G(F)$ and conversely any vertex cover of $G(F)$ of size n corresponds to a satisfying assignment in F. However, the reduction is not 'weight preserving' in the sense that a satisfying assignment of weight k can correspond to a vertex cover of size n (recall that the graph has up to $2n$ vertices). Furthermore this reduction produces a graph with the number of vertices equal to twice the number of variables and, in the parameterized setting, does not transform k into a function of k alone. Since the reduction loses track of the weight of the solution, it does not enable us to employ VERTEX COVER to solve an instance of MIN ONES 2-SAT.

1.2. Our work

In this paper, we demonstrate a simple extension of this reduction that preserves both k and n, and allows us to carry over everything we know about VERTEX COVER to the more general setting of MIN ONES 2-SAT. Thus, we have that the apparently more general problem of MIN ONES 2-SAT can be handled as easily as vertex cover, in both the optimization and parameterized settings. In particular, the problem now has a $2k - c\lg k$-variable kernel [14,15,19] (for some constant c), a 2-approximation algorithm, and FPT and exact algorithms that will run as fast as the best algorithms for the corresponding versions of the vertex cover problem, the current best being $O^*(1.2738^k)$ [4][3] and $O^*(1.2114^n)$ [2] respectively. In particular, our reduction subsumes the earlier results (2-approximation algorithms, and Nemhauser–Trotter theorem) on this problem.

[3] We use the notation $O^*()$ to "hide" functions that are polynomial in the input size.

Recently, there has been a lot of interest [6,22,19,15,13] in parameterizing VERTEX COVER above the maximum matching of the graph (and also the equivalent problem of satisfying all but at most k clauses of a 2-SAT instance). In this problem, we still seek for a vertex cover of size at most k, but like to confine the exponential explosion to $(k - m)$ where m is the size of the maximum matching in the graph (which is a lower bound for minimum vertex cover). This is addressed by using $(k - \gamma(G))$ as the parameter (instead of k), where $\gamma(G)$ is the optimum value of the LP relaxation of the integer linear programming formulation of vertex cover for G (which is lower bounded by the size of the maximum matching). We show that the optimum LP values of MIN ONES 2-SAT and the corresponding VERTEX COVER are the same. This implies a randomized kernel of size polynomial in $(k - \kappa(F))$ [13] and an $O^*(2.3146^{k-\kappa(F)})$ [15] fixed-parameter algorithm for MIN ONES 2-SAT, where $\kappa(F)$ is the optimum value of the LP relaxation of the integer programming corresponding to F.

The next section gives some definitions, particularly, in parameterized complexity. Section 3 gives the main parameter preserving reduction from MIN ONES 2-SAT to VERTEX COVER. Section 4 gives the parameter preserving reduction from the 'above guarantee' MIN ONES 2-SAT to 'above guarantee VERTEX COVER'. Finally, Section 5 gives some concluding remarks.

2. Preliminaries

A parameterized problem is denoted by a pair $(Q, k) \subseteq \Sigma^* \times \mathbb{N}$. The first component Q is a classical language, and the number k is called the parameter. Such a problem is *fixed-parameter tractable* (FPT) if there exists an algorithm that decides it in time $f(k)n^{O(1)}$ on instances of size n. A *kernelization algorithm* takes an instance (x, k) of the parameterized problem as input, and in time polynomial in $|x|$ and k, produces an equivalent instance (x', k') such that $|x'|$ is a function purely of k. The output x' is called the kernel of the problem and its size is $|x'|$. We refer the reader to [7,17] for more details on the notion of fixed-parameter tractability.

Let P be an arbitrary set, whose elements we shall refer to as *variables*. A *literal* is either a variable or its negation. An assignment for P is a function $t : P \to \{0, 1\}$. Sometimes, we also refer to an assignment setting (mapping) a variable to 'true' or 'false' when we mean to say 1 or 0 respectively.

A formula is in *conjunctive normal form* (CNF) if it is a conjunction of clauses, where a clause is a disjunction of literals. A c-SAT formula has at most c literals in any clause. The *weight* of an assignment is the number of variables that are set to one by that assignment. We refer to the problem of finding a smallest weight satisfying assignment for c-SAT formulae as MIN ONES c-SAT.

2.1. A simple FPT algorithm for weight at most k assignments

The natural parameterized version of MIN ONES c-SAT is FPT for any fixed c, when parameterized by the weight: pick a clause that contains only positive literals (as long as one exists) and branch by setting each of the variables to 1. This results in a c-way branch at each step. If in any branch at depth k, there is still a clause with all positive literals, then that branch cannot lead to an assignment with weight at most k, and hence that branch can be abandoned. If there is a formula at depth at most k where every clause contains a negative literal, then the assignment that sets all the remaining variables to 0 satisfies all such clauses. Otherwise we can conclude that the given formula has no weight k satisfying assignment. This results in an $O(c^k m)$ algorithm where m is the number of clauses in the formula.

3. Reduction of MIN ONES 2-SAT to VERTEX COVER

In this section, we present a reduction from MIN ONES 2-SAT to VERTEX COVER. Throughout, we use F to denote an instance of MIN ONES 2-SAT, and $C(F)$ denotes the set of clauses in F. Note that it is polynomial time to decide if F admits a satisfying assignment, and if F is not satisfiable, then it is a no-instance of MIN ONES 2-SAT. Therefore, we assume without loss of generality that F is a satisfiable formula.

Let $D(F)$ denote the implication graph of F, which has one vertex for every literal of F, and the directed arcs $(\bar{l}_1, l_2)$ and $(l_1, \bar{l}_2)$ for every clause $(l_1, l_2) \in C(F)$. Also, let $A(D(F))$ denote the set of arcs in $D(F)$. The implication graph of a 2-CNF formula is very well-studied, for example, see Section 1.10 in [3]. We begin by recalling Lemma 1.10.2 from [3] (the proof is reiterated here for completeness).

Lemma 1. *(See [3].) If $D(F)$ contains a path from l_1 to l_2, then, for every satisfying truth assignment t, $t(l_1) = 1$ implies that $t(l_2) = 1$.*

Proof. Observe that F contains a clause of the form $\bar{x} \vee y$ when $D(F)$ contains the arc (x, y). Further, every clause takes the value 1 under any satisfying truth assignment. Thus, by the fact that t is a satisfying truth assignment and by the definition of $D(F)$, we have that for every arc $(x, y) \in A(D(F))$, $t(x) = 1$ implies $t(y) = 1$. Now the claim follows easily by induction on the length of the shortest (l_1, l_2)-path in $D(F)$. $\square$

Let F^* be the smallest formula which contains all the clauses of F, and the clause(s) $(l_1 \vee l_2)$, for each pair of literals l_1 and l_2 such that there is a directed path from $\bar{l}_1$ to l_2 in $D(F)$. We refer to F^* as the *closure* of F. One way to compute the closure of F is to compute the transitive closure of the implication graph of F (in polynomial time, see [5]). The formula corresponding to the graph thus obtained (when treated also as an implication graph) is the closure of F. We work with the closed formula F^* in the discussion that follows.

Theorem 2. *Given a 2-CNF formula F, let F^* be the closure of F, and $(F^*)_+$ denote the set of all clauses of F^* where both literals occur positively. Let G be the graph that has one vertex for every variable in $(F^*)_+$, and $(u, v) \in E(G)$ if and only if $(u \vee v) \in C((F^*)_+)$. Then any satisfying assignment of F is a vertex cover of G and conversely any minimal vertex cover of G is a satisfying assignment of F. In particular, F has a satisfying assignment of weight at most k if and only if G has a vertex cover of size at most k.*

Proof. Consider a satisfying assignment of F. First we argue that it is a satisfying assignment of F^* as well. For, if $c = (l_1 \vee l_2)$ is in $C(F^*) \setminus C(F)$, then there is a directed path from $\bar{l}_1$ to l_2, by construction. Hence if the satisfying assignment of F sets l_1 to false, then $\bar{l}_1$ is set to true and hence by Lemma 1, l_2 is set to 1 by the assignment, thus satisfying c. Hence if F has a satisfying assignment of weight at most k, then so does F^*, and hence $(F^*)_+$, a subformula of F^*. This implies that the graph G has a vertex cover of size at most k.

Conversely let G have a minimal vertex cover K of size at most k. Let t be the truth assignment corresponding to K, i.e. let $t(x) = 1$ if $x \in K$, and $t(x) = 0$ otherwise. Clearly, t is a satisfying assignment of $(F^*)_+$ and is of weight at most k. We now show that t is indeed a satisfying assignment of F^*. The proof is by contradiction. Let us assume that F^* is not satisfied by t. This implies that there is a clause $C \in F^*$ that is not satisfied by t. Clearly, $C \notin (F^*)_+$. There are two possibilities for C: either $C = (x \vee \bar{y})$, or $C = (\bar{x} \vee \bar{y})$, where x and y are variables. In either case, we arrive at a contradiction to the assumption that t is a satisfying assignment of $(F^*)_+$.

1. $C = (x \vee \bar{y})$: Since C is falsified by t, it follows that $t(x) = 0$ (or equivalently, $t(\bar{x}) = 1$) and $t(y) = 1$. Since t is obtained from a minimal vertex cover K (that contains y as $t(y) = 1$), there is a clause $(y \vee z) \in (F^*)_+$ such that $t(z) = 0$ (as otherwise $K \setminus y$ is a vertex cover, contradicting minimality of K). By the definition of F^*, then $(x \vee z)$ is a clause in F^*, and hence in $(F^*)_+$, which is not satisfied by t, a contradiction.
2. $C = (\bar{x} \vee \bar{y})$: Since C is falsified by t, it follows that $t(x) = 1$ and $t(y) = 1$. Since t is obtained from a minimal vertex cover K, there are clauses $(y \vee z_1), (x \vee z_2) \in (F^*)_+$ such that $t(z_1) = 0, t(z_2) = 0$ (note that z_1 could be equal to z_2). It follows that $(z_1 \vee z_2)$ is a clause in F^*, and therefore it is a clause in $(F^*)_+$. Now $t(z_1) = t(z_2) = 0$, contradicting the assumption that t is a satisfying assignment of $(F^*)_+$.

Thus, F^* has a satisfying assignment of weight at most k. Since F is a subformula of F^*, it follows that so does F. □

Corollary 3. *Given a 2-CNF formula F on n variables and a positive integer k, it can be checked in time $O^*(1.2738^k)$ [4], if F admits a satisfying assignment of weight at most k. A satisfying assignment of minimum weight can be obtained in time $O^*(1.2114^n)$ and polynomial space [2], or $O^*(1.2108^n)$ and exponential space [21].*

Consider the weighted version of the problem where each variable has a non-negative real weight, and the weight of an assignment is the sum of the weights of the variables that it sets to one, and the goal is to find a minimum weight (or weight k) satisfying assignment. The reduction stated in Theorem 2 goes through for the weighted version as well. That is, the formula F has a satisfying assignment with weight at most k if and only the graph G obtained from F^* has a vertex cover of weight at most k. Thus we have

Corollary 4. *Given a 2-CNF formula F on n variables with a weight function $w : V(F) \to \mathrm{R}^+$ such that $w(v) \geqslant 1$, for all $v \in V(F)$ and a positive integer k, it can be checked in time $O^*(1.3788^k)$ [18] if F admits a satisfying assignment of weight at most k. A satisfying assignment of minimum weight can be obtained in time $O^*(1.2377^n)$ [24].*

The problem of solving a 0–1 integer program which has at most two variables per constraint with an assignment of weight at most k is known to be equivalent to MIN ONES 2-SAT. This is due to a reduction that does not increase the number of variables or the weight of the solution (Section 4, [9]). The reduction in [9] is from a more general integer program, one that assumes a bounded range (not necessarily 0–1) for each variable. However, in the general case, the number of variables created in the reduced instance is a function of the ranges. For 0–1 integer programs, the number of variables remains the same as that of the original. Thus, we also have that a binary integer program can be solved as fast as weighted vertex cover.[4]

[4] In case the coefficients for all variables in the objective function are one, then the problem may in fact be solved as fast as the unweighted vertex cover.

Corollary 5. *Consider a binary integer program where the objective function is to be minimized, and every constraint has at most two variables. Given such a program and a positive integer k, it can be checked if the optimum feasible assignment is at most k in time* $O^*(1.3788^k)$ [18], *and the optimum assignment can be obtained in time* $O^*(1.2377^n)$ [24].

4. Reduction of ABOVE GUARANTEE MIN ONES 2-SAT to ABOVE GUARANTEE VERTEX COVER

In this section, we demonstrate that the reduction presented in Section 3 (Theorem 2) also works for the above guarantee version of both the problems.

The linear program corresponding to a formula F has a variable corresponding to every variable of the formula, and a constraint corresponding to every clause. The translation of clauses to constraints is as expected, the positive literals translate to the same variables and a negative literal $\bar{x}$ would translate to $(1 - x)$. The satisfiability constraint of the clause translates to the fact that the sum of the variables is at least 1. The objective function seeks to minimize the sum of the variables. The integer programming version permits the variables to take values from the set $\{0, 1\}$, while in the “relaxed” linear program, the variables take values in the range $[0, 1]$.

Notice that the optimal value of the LP corresponding to a 2-SAT formula is a natural lower bound on the number of variables that need to be set to one to satisfy the formula. It is natural, therefore, to ask if there is a satisfying assignment of weight at most $\kappa(F) + k$ where $\kappa(F)$ denotes the value of optimum of the LP corresponding to F. We call this problem ABOVE GUARANTEE MIN ONES 2-SAT. Equivalently sometimes, we simply seek for a satisfying assignment of weight at most k, and use $k - \kappa(F)$ as the parameter.

It is known [9] that Theorem 1 generalizes to the LP corresponding to a 2-SAT formula; i.e. the LP corresponding to F admits a half-integral optimal solution, that is, one where the LP sets every variable in F to either 0, 1 or 1/2, and can be found in polynomial time. We begin by proving a version of Lemma 1 that will be useful in establishing the proof of correctness. We use the term *half-integral assignment* to refer to a setting of the variables from the range $\{0, 1/2, 1\}$, and a *satisfying* half-integral assignment is simply one that satisfies all the constraints corresponding to clauses of F.

Lemma 2. *If $D(F)$ contains a path from l_1 to l_2, then, for every satisfying half-integral assignment t of F, $t(l_2) \geqslant t(l_1)$.*

Proof. Observe that F contains a clause of the form $\bar{x} \vee y$ when $D(F)$ contains the arc (x, y). Such a clause corresponds the to constraint: $(1 - x) + y \geqslant 1$ or equivalently $y \geqslant x$. Thus, by the fact that t is a satisfying half-integral assignment and by the definition of $D(F)$, we have that for every arc $(x, y) \in A(D(F))$, $t(y) \geqslant t(x)$ and the claim follows easily by induction on the length of the shortest (l_1, l_2)-path in $D(F)$. □

We now show that the optimum value of the LP corresponding to the formula $(F^*)_+$ obtained in the previous section is exactly $\kappa(F)$, the optimum value of the LP corresponding to the 2-SAT formula.

Theorem 6. *Given a 2-CNF formula F, let F^* be the closure of F, and $(F^*)_+$ denote the set of all clauses of F^* where both literals occur positively. Let G be the LP that has one variable for every variable in $(F^*)_+$, and the constraint $u + v \geqslant 1$ if and only if $(u \vee v) \in C((F^*)_+)$. Let LP(F) be the linear programming relaxation of the integer linear programming corresponding to F. Let $\gamma(G)$ be the optimum value of the LP G, $\kappa(F)$ be the optimum value of LP(F). Then $\gamma(G) = \kappa(F)$.*

Proof. Suppose that *LP*(F) has a satisfying half-integral assignment of weight r. Then we claim that the same half-integral assignment is satisfying for F^* as well. If $c = (l_1 \vee l_2)$ is in $C(F^*) \setminus C(F)$, then there is a directed path from $\bar{l}_1$ to l_2, by construction. Hence

- if the satisfying half-integral assignment of F sets l_1 to 0, then $\bar{l}_1$ is set to 1 and hence by Lemma 2, l_2 is set to 1 by the assignment, thus satisfying the constraint corresponding to c.
- If the satisfying half-integral assignment of F sets l_1 to 1/2, then by Lemma 2, l_2 is set to at least 1/2 by the assignment, and thus the constraint corresponding to c is satisfied again.

As $(F^*)_+$ is a subformula of F^*, the same assignment is a satisfying half-integral assignment for G.

Conversely, let G have an optimal satisfying half-integral assignment t of weight r. Clearly, t is a satisfying assignment of $(F^*)_+$. Extend t to variables not in $(F^*)_+$, by setting them 0. We now show that t is indeed a satisfying assignment of F^*. The fact that it will also satisfy F is clear, since the constraints of F are a subset of the constraints in F^*.

The proof is by contradiction. Let us assume that F^* is not satisfied by t. This implies there is a clause $C \in F^*$ that is not satisfied by t. Clearly, $C \notin (F^*)_+$. There are two possibilities for C: either $C = (x \vee \bar{y})$, or $C = (\bar{x} \vee \bar{y})$, where x and y are variables. In either case, we arrive at a contradiction to the assumption that t is a satisfying assignment of $(F^*)_+$.

1. $C = (x \vee \bar{y})$: As C is falsified by t, $t(x) = 0$ or $1/2$.
   – Suppose $t(x) = 0$ (or equivalently, $t(\bar{x}) = 1$). Then $t(y) \geqslant 1/2$, and hence was set in solving G (as $t(y) > 0$). Hence, there is a clause $(y \vee z) \in (F^*)_+$ such that $t(z) \leqslant 1/2$ (else $t(y) = 0$ would be a satisfying assignment for $(F^*)_+$ of smaller weight, contradicting optimality). Hence $(x \vee z)$ would be a clause in $(F^*)_+$, but is not satisfied by t, a contradiction.
   – Suppose $t(x) = 1/2$. Then $t(y) = 1$. Again, we have that there is a clause $(y \vee z) \in (F^*)_+$ such that $t(z) = 0$. Hence, there is a clause $(x \vee z)$ in $(F^*)_+$ which is not satisfied by t, a contradiction.
2. $C = (\bar{x} \vee \bar{y})$: As C is falsified by t, one of the variables gets the value 1, and the other at least $1/2$. Without loss of generality, let $t(x) = 1$ and $t(y) \geqslant 1/2$. As t is obtained from an optimal assignment t of $(F^*)_+$, there are clauses $(y \vee z_1), (x \vee z_2) \in (F^*)_+$ such that $t(z_1) \leqslant 1/2$, $t(z_2) = 0$ (note that z_1 could be equal to z_2). Therefore, it follows that $(z_1 \vee z_2)$ is a clause in F^*, and therefore it is a clause in $(F^*)_+$, but is not satisfied by t, a contradiction.

Consequently, our assumption that t is not a satisfying assignment of F^* is wrong and hence F^* has a satisfying assignment of weight r.

Having shown that any half-integral assignment that is satisfying for F is satisfying for G and conversely, we conclude that the optimal values for both LPs are exactly the same. □

Lokshtanov et al. [15] give an algorithm to determine whether a given graph has a vertex cover of size at most k in $O^*(2.3146^{k-\gamma(G)})$ where $\gamma(G)$ is the value of the optimum of the LP relaxation of the integer programming formulation of vertex cover of G. The following corollary follows from it and Theorem 6.

Corollary 7. *Given a 2-CNF formula F on n variables and a positive integer k, it can be checked in time $O^*(2.3146^{k-\kappa(F)})$ if F admits a satisfying assignment of weight at most k where $\kappa(F)$ is the optimum value of the LP relaxation of the integer programming formulation of the 2-SAT formula F.*

A randomized polynomial kernel with one-sided error (also known as a co-RP kernel) is a randomized polynomial time algorithm that takes as input a parameterized instance (P, k) and returns an instance (Q, l), with the properties that:

1. $|Q| + l \leqslant k^{O(1)}$,
2. if (P, k) is a yes instance, then (Q, l) is a yes instance,
3. if (P, k) is a no-instance, then (Q, l) is a no instance with probability at least one-half.

Recently, Kratsch and Wahlström [13] show a polynomial co-RP kernel for the vertex cover above LP problem. From their result and Theorem 6, it follows that there is a polynomial co-RP kernel for MIN ONES 2-SAT. Formally, let F be a 2-CNF formula on n variables, and let $\kappa(F)$ denote the optimum value of the *LP* relaxation of the integer programming formulation of the 2-SAT instance F. Then, there is a polynomial time algorithm that produces a 2-CNF formula F' having $k^{O(1)}$ variables and clauses, such that: (i) there is a satisfying assignment for F' with weight at most $k + \kappa(F')$ if there is a satisfying assignment for F with weight at most $k + \kappa(F)$, and (ii) with probability at least one-half, there is no satisfying assignment for F' with weight at most $k + \kappa(F')$ if there is no satisfying assignment for F with weight at most $k + \kappa(F)$.

Corollary 8. *There is a randomized polynomial time algorithm that, given a 2-SAT formula F and an integer k, produces a 2-SAT formula F' and an integer k' such that F' has the number of variables and clauses polynomial in $k - \kappa(F)$ and if F has a satisfying assignment of weight at most k, then F' has a satisfying assignment with weight at most k' and if F has no satisfying assignment with weight at most k, then with probability at least half, F' has no satisfying assignment with weight at most k'.*

5. Concluding remarks

We have shown MIN ONES 2-SAT to be equivalent to VERTEX COVER in both the parameterized and optimization settings, by demonstrating a polynomial-time reduction from MIN ONES 2-SAT to VERTEX COVER that preserves the optimum value of both the integer and linear programming relaxation versions, and keeps the number of vertices of the graph to the number of variables in the formula. This allows us to employ the best-known algorithms and kernels for VERTEX COVER to MIN ONES 2-SAT incurring only an additional polynomial cost.

The complexity of MIN ONES c-SAT for $c > 2$ is an interesting line of research. In this case, the problem is a natural generalization of c-hitting set. While c-hitting set has a $k^{O(c)}$ kernel [1], it has been shown that a polynomial sized kernel is unlikely even for a special case of MIN ONES 3-SAT [11]. See [12] for a classification of the types of bounded variable constraints for which polynomial sized kernel is possible. Improving the obvious $O(c^k m)$ time bound (mentioned in Section 2) for the parameterized question is a natural open problem. Some progress when c is 3 has been recently reported [23].

Acknowledgement

We thank an anonymous referee for some suggestions that improved the presentation of the paper and also resulted in our observation of Corollary 8.

References

[1] Faisal N. Abu-Khzam, Kernelization algorithms for d-hitting set problems, in: Workshop on Algorithms and Data Structures (WADS), in: Lecture Notes in Computer Science, vol. 4619, Springer, 2007, pp. 434–445.
[2] Nicolas Bourgeois, Bruno Escoffier, Vangelis Th. Paschos, Johan M.M. van Rooij, Fast algorithms for max independent set, Algorithmica 62 (1–2) (2012) 382–415.
[3] Jorgen Bang-Jensen, Gregory Z. Gutin, Digraphs: Theory, Algorithms and Applications, Springer, 2008.
[4] Jianer Chen, Iyad A. Kanj, Ge Xia, Improved upper bounds for vertex cover, Theoretical Computer Science 411 (40–42) (2010) 3736–3756.
[5] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, MIT Press, Cambridge, MA, 2001.
[6] M. Cygan, M. Pilipczuk, M. Pilipczuk, J. Wojtaszczyk, On multiway cut parameterized above lower bounds, in: International Symposium on Parameterized and Exact Computation (IPEC), in: Lecture Notes in Computer Science, vol. 7112, Springer, 2012, pp. 1–12.
[7] Rod G. Downey, M.R. Fellows, Parameterized Complexity, Springer, 1999.
[8] Dan Gusfield, Leonard Pitt, A bounded approximation for the minimum cost 2-sat problem, Algorithmica 8 (1992) 103–117.
[9] D. Hochbaum, N. Meggido, J. Naor, A. Tamir, Tight bounds and 2-approximation algorithms for integer programs with two variables per inequality, Mathematical Programming 62 (1993) 69–83.
[10] Dorit S. Hochbaum (Ed.), Approximation Algorithms for NP-Hard Problems, PWS Publishing Co., Boston, MA, USA, 1997.
[11] Stefan Kratsch, Magnus Wahlström, Two edge modification problems without polynomial kernels, in: International Workshop on Parameterized and Exact Computation (IWPEC), vol. 5917, Springer, 2009, pp. 264–275.
[12] Stefan Kratsch, Magnus Wahlström, Preprocessing of min ones problems: A dichotomy, in: Automata, Languages and Programming, 37th International Colloquium (ICALP), in: Lecture Notes in Computer Science, vol. 6198, Springer, 2010, pp. 653–665.
[13] Stefan Kratsch, Magnus Wahlström, Representative sets and irrelevant vertices: New tools for kernelization, in: Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS, IEEE Computer Society, 2012, pp. 450–459.
[14] M. Lampis, A kernel of order $2k - c\log k$ for vertex cover, Information Processing Letters 111 (2011) 23–24.
[15] D. Lokshtanov, N.S. Narayanaswamy, Venkatesh Raman, M.S. Ramanujan, Saket Saurabh, Faster parameterized algorithms using linear programming, arXiv:1203.0833v1, 2012.
[16] Meena Mahajan, Venkatesh Raman, Parameterizing above guaranteed values: Maxsat and maxcut, Journal of Algorithms 31 (2) (1999) 335–354.
[17] Rolf Niedermeier, Invitation to Fixed Parameter Algorithms, Oxford University Press, USA, 2006.
[18] Rolf Niedermeier, Peter Rossmanith, On efficient fixed-parameter algorithms for weighted vertex cover, Journal of Algorithms 47 (2) (2003) 63–77.
[19] N.S. Narayanaswamy, Venkatesh Raman, M.S. Ramanujan, Saket Saurabh, LP can be a cure for parameterized problems, in: Christoph Dürr, Thomas Wilke (Eds.), 29th Symposium on Theoretical Aspects of Computer Science, STACS, in: Leibniz International Proceedings in Informatics (LIPIcs), vol. 14, 2012, pp. 338–349.
[20] George L. Nemhauser, Les E. Trotter, Vertex packings: Structural properties and algorithms, Mathematical Programming 8 (1975) 232–248.
[21] J.M. Robson, Algorithms for maximum independent sets, Journal of Algorithms 7 (3) (1986) 425–440.
[22] Venkatesh Raman, M.S. Ramanujan, Saket Saurabh, Paths, flowers and vertex covers, in: European Symposium on Algorithms, ESA, in: Lecture Notes in Computer Science, vol. 6942, Springer, 2011, pp. 382–393.
[23] Venkatesh Raman, Bal Sri Shankar, Improved fixed-parameter algorithm for the minimum weight 3-SAT problem, in: Proceedings of the 7th International Workshop on Algorithms and Computation, WALCOM, in: Lecture Notes in Computer Science, vol. 7748, 2013, pp. 265–273.
[24] Magnus Wahlström, A tighter bound for counting max-weight solutions to 2SAT instances, in: Parameterized and Exact Computation, IWPEC, in: Lecture Notes in Computer Science, vol. 5018, 2008, pp. 202–213.