

# Solving MINONES-2-SAT as fast as VERTEX COVER

Neeldhara Misra, N.S. Narayanaswamy\*, Venkatesh Raman, Bal Sri Shankar

Institute of Mathematical Sciences, Chennai, India.  
{neeldhara|swamy|vraman|balsri}@imsc.res.in

**Abstract.** The problem of finding a satisfying assignment for a 2-SAT formula that minimizes the number of variables that are set to 1 (MIN ONES 2-SAT) is NP-complete. 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  $2k$  variables kernel subsuming these 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.

## 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<sup>1</sup> at least  $k$  is a canonical complete problem for the parameterized complexity class  $W[2]$  [DF99]. If the number of variables in each clause is bounded, it is a canonical  $W[1]$ -complete problem [DF99]. 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  $O(f(k)n^{O(1)})$  on input formulas of size  $n$ .

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 independently, turns out to be fixed parameter

---

\* This work was done when the author was on sabbatical from IIT Madras

<sup>1</sup> The *weight* of an assignment is the number of variables assigned 1 by the assignment.

tractable with the weight as a parameter ([Nie06,MR99], cf. Section 2). When we restrict our attention to 2-CNF formulas (MIN ONES 2-SAT) this problem generalizes the well-studied VERTEX COVER problem. For, given a graph  $G = (V, E)$ , a satisfying assignment of weight at most  $k$  on the formula  $\bigwedge (u \vee v)$ , where the  $\bigwedge$  runs over all edges  $(u, v)$  in  $E$ , where  $u$  and  $v$  are variables corresponding to vertices  $u$  and  $v$  of  $G$ , corresponds to a vertex cover of size at most  $k$  in  $G$ . 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.

Gusfield and Pitt[GP92] considered this MIN ONES 2-SAT problem and gave a 2-approximation algorithm. The algorithm that follows a greedy approach, gives a solution whose weight is at most twice that of 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. Hochbaum et al [HMNT93] showed that the classical Nemhauser-Trotter theorem for vertex cover [NT75] 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.

There is a reduction from MIN ONES 2-SAT to VERTEX COVER, pointed out by Seffi Naor (see [Hoc97]). This reduction takes an instance  $F$  of MIN ONES 2-SAT on  $n$  variables and returns a graph  $G(F)$  that has one vertex for every literal participating in  $F$  (i.e. with up to  $2n$  vertices), and an edge between a pair of literals whenever they appear together in a clause of  $F$ , and an edge  $(x, \bar{x})$  for every variable  $x$ . Further, the reduction introduces, for every variable  $x$ , the edges  $(u, v)$  — where  $u \in N(x)$  and  $v \in N(\bar{x})$ , and  $N(l)$  is the set of all literals  $l'$  such that  $(l, l')$  is a clause of  $F$ . It can be shown that if there is a satisfying assignment of weight at most  $k$  for  $F$ , then there is a vertex cover of size  $n$  in  $G(F)$ . Conversely, a vertex cover of size  $n$  in  $G(F)$  translates to a satisfying assignment for  $F$  (i.e. that the vertex cover never needs to choose both  $x$  and  $\bar{x}$  of a variable — however, the weight of such an assignment can be as large as  $n$ ).

Observe that 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.

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 minones can be handled as easily as vertex cover, in both the optimization and parameterized settings. In particular, the problem now has a  $2k$ -variable kernel, 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

$O^*(1.27^k)$  [CKX06]<sup>2</sup> and  $O(1.2132^n)$  [KLR09]). In particular, our reduction subsumes the earlier results (2-approximation algorithms, and Nemhauser-Trotter theorem) on this problem.

## 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  $O(f(k)n^{O(1)})$  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 [DF99, Nie06] 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  $t : P \rightarrow \{0, 1\}$ . 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.

*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 of depth at most  $k$ . Notice that at the leaves, every clause has at least one negated literal and the assignment that sets all the remaining variables to 0 satisfies all such clauses. 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  $F$ . Also, let  $D(F)$  denote the implication graph of

<sup>2</sup> We use the notation  $O^*(\cdot)$  to “hide” functions that are polynomial in the variables.

$F$ , 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 [BJG08]. We begin by recalling Lemma 1.10.2 from [BJG08] (the proof is reiterated here for completeness).

**Lemma 1 ([BJG08]).** *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  $D(F)$ .

We now describe a formula that is easier to work with, because we ensure that such paths are also witnessed by edges. 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 [CLRS01]). 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 1.** *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  $F$  has a satisfying assignment of weight at most  $k$  if and only if  $G$  has a vertex cover of size at most  $k$ .*

*Proof.* Suppose that  $F$  has a satisfying assignment of weight at most  $k$ . Then it implies that the same satisfying assignment 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  $(F^*)_+$ , a sub formula of  $F^*$  has a satisfying assignment of weight at most  $k$ , which means that the graph  $G$  has a vertex cover of size at most  $k$ .

Conversely let  $G$  have a vertex cover of size  $k$ . Let  $t$  be the truth assignment corresponding to a minimal vertex cover, say  $K$ , of size at most  $k$  in  $G$ , i.e. let

$t(x) = 1$  if and only 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 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  $t(z) = 0$ .

Notice that  $D(F^*)$  has arcs from  $\bar{x}$  to  $\bar{y}$  and from  $\bar{y}$  to  $z$ , and therefore a path from  $\bar{x}$  to  $z$ . By Lemma 1, therefore,  $t(\bar{x}) = 1$  must imply  $t(z) = 1$ , 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  $z_2$ ).

As before, we observe that  $D(F^*)$  has a path from  $x$  to  $z_1$  (through  $\bar{y}$ ), and again by an application of Lemma 1, we observe that  $t(x) = 1$  implies that  $t(z_1) = 1$ , 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 at most  $k$ . Since  $F$  is a sub-formula of  $F^*$ , it follows that so does  $F$ .

**Corollary 1.** *Given a 2-CNF formula  $F$  on  $n$  variables and a positive integer  $k$ , it can be checked if  $F$  admits a satisfying assignment of weight at most  $k$  in time  $O^*(1.27^k)$  [CKX06]. A satisfying assignment of minimum weight may be obtained in time  $O(1.2132^n)$  [KLR09].*

Observe that the reduction stated in Theorem 1 is valid for 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 that the proof remains the same is easily verified.

**Corollary 2.** *Given a 2-CNF formula  $F$  on  $n$  variables with a weight function  $w : V(F) \rightarrow \mathbb{R}^+$  such that  $w(v) \geq 1$ , for all  $v \in V(F)$  and a positive integer  $k$ , it can be checked if  $F$  admits a satisfying assignment of weight at most  $k$  in time  $O^*(1.37^k)$  [NR03]. A satisfying assignment of minimum weight may be obtained in time  $O(1.32^n)$  [DJ02].*

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, [HMNT93]). The reduction in [HMNT93] 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 may be solved as fast as weighted vertex cover<sup>3</sup>.

**Corollary 3.** *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  $O^*(1.37^k)$  [NR03], and the optimum assignment may be obtained in time  $O(1.32^n)$  [DJ02].*

## 4 Concluding Remarks

We show 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 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 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 [AK07], a polynomial sized kernel is unlikely for MIN ONES  $c$ -SAT even for  $c = 3$ , as a special case of MIN ONES 3-SAT (not-1-in-3 SAT) is unlikely to have a polynomial sized kernel [KW09]. See [KWar] 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.

## References

- [AK07] Faisal N. Abu-Khzam, *Kernelization algorithms for  $d$ -hitting set problems*, WADS (Frank K. H. A. Dehne, Jörg-Rüdiger Sack, and Norbert Zeh, eds.), Lecture Notes in Computer Science, vol. 4619, Springer, 2007, pp. 434–445.
- [BJG08] Jrgen Bang-Jensen and Gregory Z. Gutin, *Digraphs: Theory, algorithms and applications*, Springer Publishing Company, Incorporated, 2008.

---

<sup>3</sup> 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.

- [CKX06] Jianer Chen, Iyad A. Kanj, and Ge Xia, *Improved parameterized upper bounds for vertex cover*, MFCS (Rastislav Kralovic and Pawel Urzyczyn, eds.), Lecture Notes in Computer Science, vol. 4162, Springer, 2006, pp. 238–249.
- [CLRS01] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*, MIT Press, Cambridge, MA, 2001.
- [DF99] Rod G. Downey and M. R. Fellows, *Parameterized complexity*, Springer, November 1999.
- [DJ02] Vilhelm Dahllöf and Peter Jonsson, *An algorithm for counting maximum weighted independent sets and its applications*, SODA, 2002, pp. 292–298.
- [GP92] Dan Gusfield and Leonard Pitt, *A bounded approximation for the minimum cost 2-sat problem*, *Algorithmica* **8** (1992), 103–117.
- [HMNT93] D. Hochbaum, N. Meggido, J. Naor, and A. Tamir, *Tight bounds and 2-approximation algorithms for integer programs with two variables per inequality*, *Mathematical Programming* **62** (1993), 69–83.
- [Hoc97] Dorit S. Hochbaum (ed.), *Approximation algorithms for NP-hard problems*, PWS Publishing Co., Boston, MA, USA, 1997.
- [KLR09] Joachim Kneis, Alexander Langer, and Peter Rossmanith, *A fine-grained analysis of a simple independent set algorithm*, IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2009) (Dagstuhl, Germany) (Ravi Kannan and K Narayan Kumar, eds.), Leibniz International Proceedings in Informatics (LIPIcs), vol. 4, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2009, pp. 287–298.
- [KW09] Stefan Kratsch and Magnus Wahlström, *Two edge modification problems without polynomial kernels*, International Workshop on Parameterized and Exact Computation (IWPEC), LNCS, 2009, pp. 264–275.
- [KWar] ———, *Preprocessing of min ones problems: A dichotomy*, 37th International Colloquium on Automata, Languages and Programming (ICALP), 2010 (to appear).
- [MR99] Meena Mahajan and Venkatesh Raman, *Parameterizing above guaranteed values: Maxsat and maxcut*, *J. Algorithms* **31** (1999), no. 2, 335–354.
- [Nie06] Rolf Niedermeier, *Invitation to fixed parameter algorithms (oxford lecture series in mathematics and its applications)*, Oxford University Press, USA, March 2006.
- [NR03] Rolf Niedermeier and Peter Rossmanith, *On efficient fixed-parameter algorithms for weighted vertex cover*, *J. Algorithms* **47** (2003), no. 2, 63–77.
- [NT75] George L. Nemhauser and Les E. Trotter, *Vertex packings: Structural properties and algorithms*, *Mathematical Programming* **8** (1975), 232–248.