Approximately: Independence Implies Vertex Cover

Sariel Har-Peled*

January 29, 2020

Abstract

We observe that a $(1-\varepsilon)$-approximation algorithm to Independent Set, that works for any induced subgraph of the input graph, can be used, via a polynomial time reduction, in provide a $(1+\varepsilon)$-approximation to Vertex Cover. This basic observation was made before, see [BHR11].

As a consequence, we get a PTAS for VC for unweighted pseudo-disks, QQPTAS for VC for unweighted axis-aligned rectangles in the plane, and QPTAS for MWVC for weighted polygons in the plane. To the best of our knowledge all these results are new.

1. Introduction

Background. Given a graph $G = (V, E)$, a ***vertex cover*** (VC) is a set $C \subseteq V$, that is adjacent to all the edges of G. The problem of computing a minimum VC is a classical problem that is **NP-Hard** [GJ90], and an easy 2-approximation algorithm is known (by computing greedily a maximal matching, and using its vertices). Dinur and Safra [DS05] showed that no approximation better than 1.3606 is possible, unless $\mathbf{P} = \mathbf{NP}$. If the unique-game conjecture is true, no approximate better than 2 is possible in the general case [KR08].

For geometric intersection graph the problem is easier. Erlebach *et al.* [EJS05] gave a PTAS for the intersection graph of weighted fat objects. Har-Peled and Quanrud [HQ17] provided a PTAS for VC for the unweighted case for low-density graphs, or polynomial expansion graphs.

The observation. Given an instance of Vertex Cover, one can reduce it in polynomial time into a dense instance, where the VC is at least half the vertices in the graph (or half the mass in the weighted case). In such a graph a $(1-\varepsilon)$-approximation to the independent set, implies readily a $(1+\varepsilon)$-approximation to VC.

Outline. For the sake of completeness, and to provide a relatively self contained description, we review this reduction (into the dense subgraph), in excruciating detail, in Section 2. We emphasize, however, that this reduction is well known – see [CC04, CC08] and references therein. The unweighted case is somewhat easier, and is described nicely in Cygan *et al.* [CFK+15, Chapter 2]. We describe the new result and its applications in Section 3.

*Department of Computer Science; University of Illinois; 201 N. Goodwin Avenue; Urbana, IL, 61801, USA; sariel@illinois.edu; http://sarielhp.org/. Work on this paper was partially supported by a NSF AF award CCF-1907400.

2. Reduction of VC to the dense case

The material covered in this section can be found in [CFK+15, CFK+15, CC04, CC08] and references therein.

Given a graph $G = (V, E)$, and a weight function $w : V \to (0, \infty)$, the ***min weight vertex cover*** (MWVC) is the problem of computing the subset $S_{\text{opt}} \subseteq V$, such that S_{opt} is adjacent to all edges in G, and $\text{vc}^*(G) = w(S_{\text{opt}}) = \sum_{v \in S_{\text{opt}}} w(v)$ is the minimum among all such sets. The complement of a vertex cover is an independent set, and in particular, we have $\text{vc}^*(G) = w(G) - \text{is}^*(G)$, where is^* is the weight of the maximum weight independent set in G, and $w(G) = \sum_{v \in V} w(v)$.

A natural starting point for approximation algorithms for VC, is to solve the associated LP:

$$\begin{aligned} \min \quad & \textstyle\sum_{v \in V} w(v) x_v & & \\ \text{s.t.} \quad & x_u + x_v \geq 1 & \forall uv \in E(G) & \qquad \text{(LPVC)} \\ & x_v \geq 0 & \forall v \in V. & \end{aligned}$$

For an assignment $\mathsf{z} = (z_v)_{v \in V}$, let $\nu(\mathsf{z}) = \sum_{v \in V} w(v) z_v$ denote the ***value*** of z.

Lemma 2.1. *There is an optimal solution for* (LPVC) *that is half-integral. That is, for any $v \in V$, we have $x_v \in \{0, 1/2, 1\}$.*

Proof: Consider an optimal solution $\mathsf{x} = (x_v)_{v \in V}$ for (LPVC), Let $L = \{v \in V \mid x_v \in (0, 1/2)\}$ and $H = \{v \in V \mid x_v \in (1/2, 1)\}$. If $w(L) < w(H)$, then consider the assignment

$$y_v = \begin{cases} x_v - \varepsilon & v \in H \\ x_v + \varepsilon & v \in L \\ x_v & \text{otherwise,} \end{cases}$$

where $\varepsilon = \min_{v \in H \cup L} |x_v - 1/2|$. For an edge uv, with $u \in L$ and $v \in R$, we have $y_u + y_v = x_u + \varepsilon + x_v - \varepsilon = x_u + x_v \geq 1$. It is easy to verify, in a similar fashion, that $\mathsf{y} = (y_v)_{v \in V}$ is a feasible solution for (LPVC). Furthermore, since $w(L) < w(H)$, we have

$$\nu(\mathsf{y}) = \sum_{v \in V} w(v) x_v - \varepsilon w(H) + \varepsilon w(L) = \nu(\mathsf{x}) + \varepsilon(w(L) - w(H)) < \nu(\mathsf{x}),$$

which is impossible, by the optimality of x. A similar argument applies if $w(L) > w(H)$. As such, it must be that $w(L) = w(H)$. But then, the solution y is also optimal, and it has one more variable assigned a value of $1/2$ then x. Repeating this argument now to y till both L and R are empty implies the claim. ■

Lemma 2.2. *An optimal half-integral solution to the linear program (LPVC) can be computed in $O(n^3)$ time, where $n = |V|$.*

Proof: Let K be the bipartite graph over the bipartition $U_1 = \{u_1 \mid u \in V\}$ and $U_2 = \{u_2 \mid u \in V\}$. Let

$$E(K) = \{u_1 v_2,\ u_2 v_1 \mid uv \in E(G)\}.$$

For any vertex $v_i \in U_1 \cup U_2$, its weight is its original weight $w(v)$.

There is an associated network flow instance – adding a source vertex s, and a sink vertex t. Here, the source vertex s is connected to all the vertices in U_1. An edge (s, u_1), for $u_1 \in U_1$, has capacity $w(u)$.

All the edges of K or oriented from U_1 to U_2, with infinite capacity, and an edge u_2t, for all $u_2 \in U_2$, has capacity $w(u_2) = w(u)$. Let N denote the resulting instance of network flow. It is easy to verify that a s-t min-cut in N corresponds to a minimum weight vertex cover in K, and vice versa. As such, compute a max-flow in N, and let f denote this flow. Given f, the corresponding s-t min-cut can be computed in linear time from the residual network flow of f (i.e., min-cut max-flow theorem). This cut then can be readily converted into the desired minimum vertex cover S_{opt} in K. Observe that $w(S_{\mathrm{opt}}) = |f|$, where $|f|$ is the value of the flow of f. The max-flow computation in this case can be done in $O(n^3)$ time using known algorithms [CLRS01].

For a vertex $v \in V(K)$, let $\chi(v) = 1$ if $v \in S_{\mathrm{opt}}$, and zero otherwise. We set $x_v = \big(\chi(v_1) + \chi(v_2)\big)/2$, for all $v \in V$. Let $\mathsf{x} = (x_v)_{v \in V}$. We have

$$\nu(\mathsf{x}) = \sum_{v \in V} w(v) x_v = \frac{w(S_{\mathrm{opt}})}{2}. \tag{1}$$

Consider an edge $uv \in E(G)$. Observe that

$$x_u + x_v = \frac{\chi(u_1) + \chi(u_2) + \chi(v_1) + \chi(v_2)}{2} = \frac{\chi(u_1) + \chi(v_2)}{2} + \frac{\chi(u_2) + \chi(v_1)}{2} \geq \frac{1}{2} + \frac{1}{2} = 1,$$

since S_{opt} is a vertex cover of K. So $\mathsf{x} = (x_v)_{v \in V}$ is a feasible assignment for (LPVC) (and it also half-integral).

Consider an optimal half-integral assignment $\mathsf{y} = (y_v)_{v \in V}$ for LPVC, which exists by Lemma 2.1.. This induces a natural vertex cover for K. Indeed, if $x_v = 1/2$ we add v_1 to a set T. Similarly, if $x_v = 1$, we add v_1 and v_2 to T. We claim that T is a vertex cover for K. Indeed, for any edge $uv \in E(G)$, we have that $x_u + x_v \geq 1$. If $x_u = 1$ then $u_1, u_2 \in T$ and these two vertices cover the edges u_1v_2 and u_2v_1. A similar argument applies if $x_v = 1$. The remaining possibility is that $x_u = x_v = 1/2$, but then $u_1, v_1 \in T$. Which implies that T covers the edges u_1v_2 and u_2v_1. As such, all the edges of K are covered by T.

Observe that

$$2\nu(\mathsf{y}) = 2\sum_{v \in V} w(v) y_v = w(T) \geq w(S_{\mathrm{opt}}) = 2\nu(\mathsf{x}),$$

by Eq. (1). Namely $\nu(\mathsf{y}) \geq \nu(\mathsf{x})$. We conclude that x is an optimal solution for (LPVC), and it is also half-integral. ■

Given an optimal half-integral solution $\mathsf{x} = (x_v)_{v \in V}$ for (LPVC), one can partition the vertices into three sets:

(i) $V_0 = \{v \in V \mid x_v = 0\}$.

(ii) $V_{1/2} = \{v \in V \mid x_v = 1/2\}$.

(iii) $V_1 = \{v \in V \mid x_v = 1\}$.

A few observations about these sets. The set V_0 is an independent set in G. Indeed, an edge $uv \in E$ with an endpoint $u \in V_0$ and an endpoint in $v \in V_0 \cup V_{1/2}$ has $x_u + x_v \leq 0 + 1/2 = 1/2$, which is impossible. As such, for a vertex $v \in V_0$, any adjacent edge $vu \in E$, must have $u \in V_1$. Namely, there is no edge between a vertex of V_0 and a vertex $V_{1/2}$. This is known as a ***crown decomposition***, with $C = V_0$ being the "crown", $R = V_{1/2}$ being the "body", and $H = V_1$ be the "head".

We need the weighted version of the Nemhauser-Trotter theorem, which we prove next.

Theorem 2.3 (Nemhauser-Trotter). *Let G be a graph with weights $w : V \to (0, \infty)$. There is a minimum weight vertex cover S of G, such that $V_1 \subseteq S \subseteq V_1 \cup V_{1/2}$.*

Proof: Let S_{opt} be the optimal vertex cover, and let $S = (S_{\mathrm{opt}} \setminus V_0) \cup V_1$. Clearly, S is a vertex cover, as all the edges adjacent to V_0 are covered by V_1. Since $V_0 \cup V_{1/2} \cup V_1 = V$, we have $S = V_1 \cup S \subseteq V_{1/2} \cup V_1$.

Assume, for the sake of contradiction, that $w(S) > w(S_{\mathrm{opt}})$. We have

$$
\begin{aligned}
& w(S) = w(S_{\mathrm{opt}}) - w\big(V_0 \cap S_{\mathrm{opt}}\big) + w\big(V_1 \setminus S_{\mathrm{opt}}\big) > w(S_{\mathrm{opt}}) \\
\iff\ & w(V_0 \cap S_{\mathrm{opt}}) - w(V_1 \setminus S_{\mathrm{opt}}) < 0.
\end{aligned}
$$

Let $\varepsilon < 1/2$ be some constant, and for any $v \in V$, let

$$
y_v = \begin{cases} 1 - \varepsilon & v \in V_1 \setminus S_{\mathrm{opt}} \\ \varepsilon & v \in V_0 \cap S_{\mathrm{opt}} \\ x_v & \text{otherwise.} \end{cases}
$$

An easy case analysis shows that the $\mathsf{y} = (y_v)_{v \in V}$ is a feasible solution. Indeed, we need to consider only edges uv with at least one endpoint (say u) in $V_1 \setminus S_{\mathrm{opt}}$, (as these are the only edges that lost value). There are the following cases to consider:

(i) $v \in V_1$: We have $y_u + y_v \geq 1 - \varepsilon + 1 - \varepsilon \geq 1$.

(ii) $v \in V_{1/2}$: We have $y_u + y_v = 1 - \varepsilon + 1/2 \geq 1$.

(iii) $v \in V_0 \cap S_{\mathrm{opt}}$: We have $y_u + y_v = 1 - \varepsilon + \varepsilon = 1$.

(iv) $v \in V_0 \setminus S_{\mathrm{opt}}$: Both endpoints of uv are outside S_{opt}, which is impossible.

Observe that $\nu(\mathsf{y}) = \sum_{v \in V} y_v = \nu(\mathsf{x}) + \varepsilon w(V_0 \cap S_{\mathrm{opt}}) - \varepsilon w(V_1 \setminus S_{\mathrm{opt}}) < \nu(\mathsf{x})$, which is a contradiction to the optimality of x. ■

Lemma 2.4. *Let $K = G[V_{1/2}]$ be the induced subgraph of G over $V_{1/2}$. Consider any minimum vertex cover C of $G[V_{1/2}]$. Then $C \cup V_1$ is a minimum weight vertex cover of G.*

Proof: Let $H = G[V_0 \cup V_{1/2}]$. Consider any minimum vertex cover C to the graph H and observe that $C \cup V_1$ is a vertex cover for G. Similarly, given a minimum vertex cover S_{opt} for G, consider the equally priced vertex cover $S = (S_{\mathrm{opt}} \setminus V_0) \cup V_1$ (see Theorem 2.3), and observe that S is a vertex cover of G and $S \setminus V_1$ is a vertex cover for H. We conclude that computing the minimum vertex cover for G, is equivalent to computing the vertex cover for H. Since V_0 is an independent set in G, and it is connected to only vertices of V_1, it follows that V_0 is a set of isolated vertices in H. As such, it is sufficient to compute a minimum vertex cover for $G[V_{1/2}]$. ■

Observe that x induces a valid optimal fraction solution for the induced subgraph graph $G[V_{1/2}]$.

Lemma 2.5. *Consider any fractional optimal solution y for the vertex cover of the graph $G[V_{1/2}]$. We have that $\nu(\mathsf{y}) = \nu(\mathsf{x}) - w(V_1) = \sum_{v \in V_{1/2}} w(v)/2 = w(V_{1/2})/2$.*

Proof: The assignment y, can be extended into a fractional solution for G, by setting $y_v = 1$ if $v \in V_1$, and zero otherwise. Similarly, x when restricted to $V_{1/2}$ is a valid VC solution for $G[V_{1/2}]$. This readily implies the claim, as x is the optimal solution for (LPVC). ■

3. The result and some applications

For a weighted graph G, let $\text{vc}^*(G)$ denote the weight of the minimum weight vertex cover of G. Similarly, let $\text{is}^*(G)$ the weight of the maximum weight independent set in G.

Theorem 3.1. *Let $G = (V, E)$ be a graph, with weights on the vertices, and assume that we are given an algorithm* `algIS` *that can compute, in $T(n)$ time, an independent set $I \subseteq V$, such that $w(I) \geq (1-\varepsilon)\text{is}^*(G)$. Furthermore, assume that this algorithm works for any induced subgraph of G. Then, one can compute, in $O(n^3 + T(n))$ time, a vertex cover C of G, such that $w(C) \leq (1+\varepsilon)\text{vc}^*(G)$.*

Proof: Using the algorithm of Lemma 2.4, compute a partition of the vertices of G into the three sets $V_0, V_{1/2}, V_1$. This reduces G into an induced subgraph $K = G[V_{1/2}]$, such that it is enough to solve the problem on K. Importantly, for the graph K, we have the property that the optimal LP solution (for the relaxation of the LP) assigns all vertices value $1/2$. That is $\text{vc}^* = \text{vc}^*(K) \geq w/2$, where $w = \sum_{v \in V(K)} w(v)$.

In the following, let $\text{is}^* = \text{is}^*(K)$. Compute an independent set I in K using `algIS`. We have that

$$w(I) \geq (1-\varepsilon)\text{is}^* = (1-\varepsilon)(w - \text{vc}^*).$$

The set $C = V(K) - I$ is a vertex cover, and we have

$$\begin{aligned} w(C) = w - w(I) &\leq w - (1-\varepsilon)(w - \text{vc}^*) = \varepsilon w + (1-\varepsilon)\text{vc}^* \\ &\leq 2\varepsilon\, \text{vc}^* + (1-\varepsilon)\text{vc}^* = (1+\varepsilon)\text{vc}^*, \end{aligned}$$

since $\text{vc}^* \geq w/2$. We have that $C \cup V_1$ is a vertex cover for the original graph. Furthermore, by Lemma 2.4, we have

$$w(C \cup V_1) = w(C) + w(V_1) \leq (1+\varepsilon)\text{vc}^* + w(V_1) \leq (1+\varepsilon)(w(V_1) + \text{vc}^*) = (1+\varepsilon)\text{vc}^*(G),$$

since $\text{vc}^*(G) = \text{vc}^* + w(V_1)$ by Lemma 2.4. ■

3.1. Applications

For several cases, efficient $(1+\varepsilon)$-approximations algorithms are known:

(i) A PTAS for the unweighted pseudo-disks [CH12].
(ii) A QQPTAS is for unweighted rectangles in the plane [CE16].
(iii) A QPTAS for weighted simple polygons in the plane [AHW19].

Plugging these results into Theorem 3.1, implies the following.

Theorem 3.2. *We have the following:*

(i) For the intersection graph of n unweighted pseudo-disk in the plane, a $(1+\varepsilon)$-approximation to the minimum vertex cover can be computed in $n^{O(1/\varepsilon^2)}$ time.

(ii) For the intersection graph of n unweighted axis-aligned rectangles in the plane, a $(1+\varepsilon)$-approximation to the minimum vertex cover can be computed in $n^{O(\text{poly}(\log\log n, 1/\varepsilon))}$ time.

(iii) For the intersection graph of n weighted simple polygons in the plane, a $(1+\varepsilon)$-approximation to the minimum weight vertex-cover can be computed in $n^{O(\text{poly}(\log n, 1/\varepsilon))}$ time.

Acknowledgments. The author thanks Chandra Chekuri for useful discussions.

References

[AHW19] A. Adamaszek, S. Har-Peled, and A. Wiese. *Approximation schemes for independent set and sparse subsets of polygons*. *J. Assoc. Comput. Mach.*, 66(4): 29:1–29:40, 2019.

[BHR11] R. Bar-Yehuda, D. Hermelin, and D. Rawitz. *Minimum vertex cover in rectangle graphs*. *Comput. Geom.*, 44(6-7): 356–364, 2011.

[CC04] M. Chlebík and J. Chlebíková. *Improvement of Nemhauser-Trotter theorem and its applications in parametrized complexity*. *Proc. 9th Scand. Workshop Algorithm Theory* (SWAT), vol. 3111. 174–186, 2004.

[CC08] M. Chlebík and J. Chlebíková. *Crown reductions for the minimum weighted vertex cover problem*. *Discrete Applied Mathematics*, 156(3): 292–312, 2008.

[CE16] J. Chuzhoy and A. Ene. *On approximating maximum independent set of rectangles*. *Proc. 57th Annu. IEEE Sympos. Found. Comput. Sci.* (FOCS), 820–829, 2016.

[CFK+15] M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized algorithms*. Springer, 2015.

[CH12] T. M. Chan and S. Har-Peled. *Approximation algorithms for maximum independent set of pseudo-disks*. *Discrete Comput. Geom.*, 48(2): 373–392, 2012.

[CLRS01] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to algorithms*. MIT Press / McGraw-Hill, 2001.

[DS05] I. Dinur and S. Safra. *On the hardness of approximating vertex cover*. *Annals of Mathematics*, 162(1): 439–485, 2005.

[EJS05] T. Erlebach, K. Jansen, and E. Seidel. *Polynomial-time approximation schemes for geometric intersection graphs*. *SIAM J. Comput.*, 34(6): 1302–1323, 2005.

[GJ90] M. R. Garey and D. S. Johnson. *Computers and intractability; a guide to the theory of NP-completeness*. W. H. Freeman & Co., 1990.

[HQ17] S. Har-Peled and K. Quanrud. *Approximation algorithms for polynomial-expansion and low-density graphs*. *SIAM J. Comput.*, 46(6): 1712–1744, 2017.

[KR08] S. Khot and O. Regev. *Vertex cover might be hard to approximate to within* $2-\varepsilon$. *J. Comput. Sys. Sci.*, 74(3): 335–349, 2008.