An Improved Approximation Algorithm For Vertex Cover with Hard Capacities

(Extended Abstract)

Rajiv Gandhi[*] Eran Halperin[†] Samir Khuller[‡] Guy Kortsarz[§]
Aravind Srinivasan[¶]

Abstract

In this paper we study the capacitated vertex cover problem, a generalization of the well-known vertex cover problem. Given a graph $G = (V, E)$, the goal is to cover all the edges by picking a minimum cover using the vertices. When we pick a vertex, we cover up to a pre-specified number of edges incident on this vertex (its capacity). The problem is clearly NP-hard as it generalizes the well-known vertex cover problem. Previously, 2-approximation algorithms were developed with the assumption that multiple copies of a vertex may be chosen in the cover. If we are allowed to pick at most a given number of copies of each vertex, then the problem is significantly harder to solve. Chuzhoy and Naor (*Proc. IEEE Symposium on Foundations of Computer Science, 481–489, 2002*) have recently shown that the weighted version of this problem is at least as hard as set cover; they have also developed a 3-approximation algorithm for the unweighted version. We give a 2-approximation algorithm for the unweighted version, improving the Chuzhoy-Naor bound of 3 and matching (up to lower-order terms) the best approximation ratio known for the vertex cover problem.

1 Introduction

The capacitated vertex cover problem can be described as follows. Let $G = (V, E)$ be an undirected graph with vertex set $V = \{1, ..., n\}$ and edge set E. Suppose that w_v denotes the weight of vertex v and k_v denotes the capacity of vertex v (we assume that k_v is an integer). A *capacitated vertex cover* is a function that determines a value $x_v \in \{0, 1, \ldots, b_v\}$, $\forall v \in V$ such that there exists an orientation of the edges of G in which the number of edges directed into vertex $v \in V$ is at most $k_v x_v$. (These edges are said to be *covered* by or *assigned* to v.) The *weight* of the cover is $\sum_{v \in V} x_v w_v$. The MINIMUM CAPACITATED VERTEX COVER problem is that of computing a minimum

---

[*] Department of Computer Science, University of Maryland, College Park, MD 20742. Research supported by NSF Award CCR-9820965. E-mail: `gandhi@cs.umd.edu`.

[†] International Computer Science Institute, Berkeley, CA 94704 and Computer Science Division, University of California, berkeley, CA 94720. Supported in part by NSF grants CCR-9820951 and CCR-0121555 and DARPA cooperative agreement F30602-00-2-0601. E-mail: `eran@cs.berkeley.edu`.

[‡] Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742. This research was supported by NSF Award CCR-9820965 and CCR-0113192. E-mail: `samir@cs.umd.edu`.

[§] Department of Computer Science, Rutgers University, Camden, NJ 08102. E-mail: `guyk@camden.rutgers.edu`.

[¶] Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742. Supported in part by NSF Award CCR-0208005. E-mail: `srin@cs.umd.edu`.

weight capacitated cover. The problem generalizes the MINIMUM WEIGHT VERTEX COVER problem which can be obtained by setting $k_v = |V| - 1$ for every $v \in V$. The main difference is that in vertex cover, by picking a node v in the cover we can cover all edges incident to v, in this problem we can only cover a subset of at most k_v edges incident to node v.

Guha *et al.* [8] studied the version of the problem in which b_v is unbounded. They obtain a 2-approximation algorithm using the primal-dual method. They also gave a 4-approximate solution using LP-rounding. Gandhi *et al.* [7] gave a 2-approximate solution using LP-rounding.

The problem becomes significantly harder when b_v is bounded for each vertex $v \in V$. For arbitrary weights on the vertices the problem is at least as hard as the set cover problem. For the case when $w_v = 1$, for all $v \in V$, Chuzhoy and Naor [5] give a nice 3-approximation algorithm for this problem. Their algorithm uses randomized LP-rounding with alterations. In this paper, we modify the algorithm of Chuzhoy and Naor in two crucial ways to obtain a 2-approximate solution. This is in a sense the "best ratio" possible at the moment, as 2 is also the best ratio known for the simpler vertex-cover problem. We add a pre-processing step in which we make certain capacity-1 vertices ineffective by making their capacities 0. We also modify their alteration step in an important way that helps us to bound the cost of the alteration step in a better way and changes the algorithm.

Related work: The best-known approximation algorithms for the vertex cover problem achieve an approximation ratio of $(2 - o(1))$ for arbitrary graphs [1, 10, 11]. A nice overview of the work on this problem is presented in [13].

The vertex cover is a special case of the set-cover problem that requires to select a minimum number (or minimum cost) collection of subsets that cover the entire universe. The set-cover with hard capacities problem generalizes the set-cover problem in that a set has a capacity bound on the number of elements it can cover. In a seminal paper [9], Johnson gave the first (greedy) logarithmic ratio approximation for the unweighted uncapacitated set cover problem. This was generalized in [3] to the weighted uncapacitated case, and further generalized in [6] to approximating with logarithmic ratio the integer linear program min $c \cdot x$ subject to $Ax \geq b$ with all the entries in A nonnegative. A much more general result is given in [18] giving a logarithmic ratio approximation algorithm for submodular cover problems. Both the vertex cover problem with hard capacities, and set cover problem with hard capacities are an example of a submodular cover problem. Hence [18] gave the first nontrivial approximation for both problems. See also [2] for a generalization of the method including, e.g., generalization of the set-cover problem with hard capacities problem, facility location problems under flow constraints and the $2-$layered facility location problem (without triangle inequality) under hard capacity constrains.

Indeed, closely related problem to set cover with hard capacities, is facility location with hard capacities. In this problem, we are given a set of facilities F and a set of clients C. There is a cost function $d : L \to F$ which defines the cost of assigning a client to a facility. Each facility $f \in F$ has a cost w_f, a bound b_f denoting the number of available copies of f and capacity k_f denoting the maximum number of clients that can be assigned to an open facility. Each client i has demand g_i. The goal is to open facilities so that each client can be assigned to some open facility. The objective is to minimize the sum of cost of open facilities and the cost of assigning the clients to them. A logarithmic greedy approximation problem for the uncapacitated case appears in [12] and for the capacitated case and some generalizations in [2]. Slightly improved (still logarithmic) bounds for the uncapacitated case is given in [19] using randomized methods.

There has been a lot of work on metric facility location (see [17] for details). For the metric facility location problem with hard capacities, Pál, Tardos and Wexler [16] gave a $(9+\epsilon)$-approximation algorithm using local search.

Research has also been conducted on the multi-set multi-cover problem. In this problem, the input sets are actually multi-sets, i.e., an element can appear in a set more than once. The problem with unbounded set capacities can be defined as the following IP: $\min\{w^T x | Ax \geq d, 0 \leq x \leq b, x \in Z\}$. The LP has an unbounded Integrality gap. Dobson [6] gave a greedy algorithm achieving a guarantee of $H(\max_{1 \leq j \leq n} A_{ij})$. Recently, Carr *et al.* [4] gave a p-approximation algorithm, where p denotes the maximum number of variables in any constraint. Their algorithm is based on a stronger LP-relaxation. Kolliopoulos and Young [14] obtained an $O(\log n)$ approximation algorithm.

Remark: We will solve the special case of the vertex cover problem in which at most one copy of each vertex can be used. As in [5], our algorithm can be easily extended to the general case where multiple copies of each vertex can be used.

2 IP Formulation and Relaxation

A linear integer program (IP) of the problem can be written as follows.

In this formulation, $y_{ev} = 1$ denotes that the edge $e \in E$ is covered by vertex v. Clearly, the values of x in a feasible solution correspond to a capacitated cover. While we do not really need the constraint $x_v \geq y_{ev}$ $v \in e \in E$ for the IP formulation, this constraint will play an important role in the relaxation. (In fact, without this constraint there is a large Integrality gap between the best fractional and integral solutions). For any vertex v, let $E(v)$ denote the set of edges incident on v.

$$\begin{array}{ll} \text{Minimize } \sum_v x_v & \\ y_{eu} + y_{ev} = 1 & e = \{u, v\} \in E, \\ k_v x_v - \sum\limits_{e \in E(v)} y_{ev} \geq 0 & v \in V, \\ x_v \geq y_{ev} & v \in e \in E, \\ y_{ev} \in \{0, 1\} & v \in e \in E, \\ x_v \in \{0, 1\} & v \in V. \end{array} \tag{1}$$

In the relaxation to a linear program, we restrict $y_{ev} \geq 0$ and $0 \leq x_v \leq 1$.

3 Algorithm

Our algorithm differs from the Chuzhoy-Naor algorithm in the following two ways. We perform a pre-processing step (Step 1) in which we make some of the capacity-1 vertices ineffective by making their capacities 0. Our alteration step (Step 5) is also different than the alteration step used in the Chuzhoy-Naor algorithm. Both these changes are crucial to our analysis. Let (x', y') be a solution in which x' is a binary vector and y' is fractional. Once we have such a solution, we can convert it to a solution (x', y'') in which y'' is integral (Step 6).

1. **Pre-Processing.** Keep "removing" capacity 1 vertices (make their capacity 0) from the graph until we have a graph in which removing any capacity 1 vertex will result in an infeasible solution. Include the remaining capacity 1 vertices in the cover (add the "$x_v = 1$" constraint in the LP for each such vertex v). Checking whether a graph, $G = (V, E)$, has a feasible solution or not can be done as follows. Let $B = (A_1, A_2, F)$ be a bipartite graph in which each node in A_1 represents an edge in E and the vertex set $A_2 = V$. An edge $(a, b) \in F$ iff in G, edge a is incident on vertex b. Construct a flow network in which the source is connected to all vertices in A_1 and each vertex in A_2 is connected to the sink. The capacities of the edges in F is 1. The capacities of the edges emanating from the source are all 1. The capacity of an edge from any node $v \in A_2$ to the sink is k_v. Now, G has a feasible solution iff the maximum flow from the source to the sink is $|E|$.

2. **LP Solution.** Solve the LP relaxation (that has the additional constraint $x_v = 1$ for each capacity-1 vertex v that survived the pre-processing step) optimally. To facilitate the discussion of the remainder of the algorithm let us introduce some notation.

   $U = \{u | x_u \geq 1/2\}$.
   $\overline{U} = V \setminus U$.
   $E' = \{(u, v) | u \in U, v \in \overline{U}\}$.
   $\forall u \in V, E'(u) = E' \cap E(u)$ and $d_u = |E'(u)|$.
   $\forall u \in U, x_u = 1 - \epsilon_u, 0 \leq \epsilon_u \leq 1/2$.

3. **Partial Cover.** Include all vertices of U in the cover, i.e., $\forall u \in U, x'_u = 1$. Note that all capacity-1 vertices belong to U. For any edge $e = (u, v) \in E \setminus E'$, set $y'_{eu} = y_{eu}$ and $y'_{ev} = y_{ev}$. The contribution of $u \in U$ towards covering edge $e = (u, v) \in E'(u)$ is at least $y'_{eu} = y_{eu}/x_u \geq (1 - y_{ev})/(1 - \epsilon_u)$. For each $e = (u, v) \in E'(u)$, let $h_{ev} = 1 - (1 - y_{ev})/(1 - \epsilon_u) = (y_{ev} - \epsilon_u)/(1 - \epsilon_u)$. To cover all the edges in $E'(u)$ fractionally, we are going to need an additional coverage of $h_u = \sum_{e=(u,v) \in E'(u)} h_{ev}$. In the following steps we will get the necessary additional coverage from vertices in $\overline{U}$. Note that there are no edges within $\overline{U}$.

4. **Randomized Rounding.** Round each vertex $v \in \overline{U}$ to 1 *independently* with probability $2x_v$. Let I be the set of vertices that are rounded to 1 in this step. For each edge $e = (u, v) \in E'$ such that $v \in I$, let $y'_{ev} = y_{ev}/x_v$ be the contribution of v towards covering e. By constraint (1), $\sum_{e \in E(v)} y_{ev}/x_v = \sum_{e \in E(v)} y'_{ev} \leq k_v$.

5. **Alteration.** Let P be the set of vertices in U that still need some help from vertices in $\overline{U}$, i.e., $P = \{u \in U | \sum_{e=(u,v), v \in I} y'_{ev} < h(u)\}$. In this step, we will choose a set of vertices $I' \subseteq \overline{U} \setminus I$, such that $\forall u \in P, \sum_{e=(u,v), v \in I \cup I'} y'_{ev} \geq h(u)$, where for each vertex $v \in I'$, y'_{ev} is set as shown in step (c) below. For each vertex $u \in P$, we define a set of vertices *helper*(u). Each vertex in *helper*(u) contributes towards h_u. Each vertex in I' belongs to exactly one such set. Initially, $I' \leftarrow \emptyset$ and *helper*$(u) \leftarrow \emptyset, \forall u \in P$. We perform the following steps until the set P is empty.

   (a) Pick a vertex $u \in P$.
   (b) Consider any edge (u, v) such that $v \in \overline{U} \setminus (I \cup I')$. *helper*$(u) \leftarrow$ *helper*$(u) \cup \{v\}$. $I' \leftarrow I' \cup \{v\}$. Let $P'_v = \{w \in P | w \neq u, e' = (w, v) \in E'\}$.
   (c) For each $w \in P'_v$ and $e' = (w, v)$, set $y'_{e'v} = y_{e'v}$ and set $y'_{e'w} = 1 - y'_{e'v}$. Set y'_{ev}, where $e = (u, v)$, to be the minimum of 1 and the remaining capacity of v. Set $y'_{eu} = 1 - y'_{ev}$.

   (d) For each vertex $w \in P'_v$, if $\sum_{e=(w,a),a \in I \cup I'} y'_{ea} \geq h_w$ remove w from P. For each edge $f = (w, b) \in E'$ such that $b \notin I \cup I'$, set $y'_{fb} = 0$ and $y'_{fw} = 1$.

   (e) Remove u from P iff $\sum_{e=(u,a),a \in I \cup I'} y'_{ea} \geq h_u$.

   Once P is empty, we have a feasible solution (x', y') in which x' is integral and y' may be fractional.

6. **Integral Solution.** At this point x' is a binary vector but y' is fractional. This can be converted to an integral solution using the Integrality of flows property on a flow network. The flow network is exactly the same as the one constructed in Step 1 with the difference that the capacity of an edge going from a node representing $v \in V$, to the sink is $k_v x'_v$.

4 Analysis

In Step 4 of the algorithm we choose the set of vertices I' and include them as part of our cover. We have to account for the cost of these vertices. Note that for each vertex $v \in I'$ there is exactly one vertex $u \in P$, such that $v \in helper(u)$. We will charge to u the cost of adding v to our solution. Note that in the LP solution the cost of vertex u is $1 - \epsilon_u$. In our solution, vertex $u \in U$ pays for itself and for the vertices in $helper(u)$. We will show that the total expected charge on u due to vertices in $helper(u)$ is at most $1 - 2\epsilon_u$. Thus, the total expected cost of vertex u is $2 - 2\epsilon_u = 2x_u$. Also, the total expected size of I is $\sum_{v \in \overline{U}} x_v$. Thus we will get that we have a 2-approximation in expectation, by using the linearity of expectation.

Our primary goal will be to show that for any $u \in U$, the total expected charge on u due to vertices in $helper(u)$ is at most $1 - 2\epsilon_u$. Before doing so, we will first show that our preprocessing step is justifiable:

Lemma 4.1 *Let R be the set of vertices of capacity 1 removed from a graph G_o in the pre-processing step (Step 1). Let G_n be the new graph that has the same vertices and edges as G_o except that the capacities of the vertices in R is reduced to 0. Let $OPT(G_o)$ and $OPT(G_n)$ represent the optimal solutions in G_o and G_n respectively. Then $OPT(G_o) = OPT(G_n)$. This implies that the LP solution to G_n is a lower bound on $OPT(G_o)$.*

Proof Let $OPT(G_o)$ be an optimal solution to G_o that uses minimum number of vertices from R. If $OPT(G_o) \cap R = \emptyset$ then the claim follows trivially. Now consider the case when $OPT(G_o) \cap R \neq \emptyset$. Let $v \in OPT(G_o) \cap R$. Construct a directed graph H having the same vertex set as G_o. Include an edge (a, b) in H iff edge (a, b) in G_o is covered by a in $OPT(G_o)$ and by b in $OPT(G_n)$. H may contain some cycles. Since v has in-degree zero in H, v cannot be part of any cycle. Contract every cycle of H. Now consider a maximal path, Q, starting from v. Let w be the last vertex in the path. Note that w does not have any outgoing edges, otherwise Q is not maximal. Consider the solution $OPT(G_o) \setminus \{v\} \cup \{w\}$ in which the edges of Q have the same assignment as in $OPT(G_n)$. We will now show that this new assignment does not violate capacity constraints of any of the vertices. The only vertices that are affected are the vertices in Q. The assignment of edges to all other vertices remain the same as in $OPT(G_o)$. In H, since w has one incoming edge and no outgoing edges, w covers one more edge in $OPT(G_n)$ than it covers in $OPT(G_o)$. Since $w \notin R$, the capacity of w is the same in G_o and in G_n. Thus w covers at most $k_w - 1$ edges in $OPT(G_o)$. Thus in $OPT(G_o)$, w has a spare capacity of at least 1 that it uses to cover its incoming edge in Q. Every other vertex whose covering is different than in $OPT(G_o)$ is an internal vertex of Q. Each such vertex uncovers one edge (outgoing edge in Q) and covers a new edge (incoming edge in Q), hence its capacity constraints are not affected. This cost of this solution is the same as $OPT(G_o)$ and it uses one fewer vertex from R, thus contradicting the assumption that $OPT(G_o)$ used minimum number of vertices from R. ■

Lemma 4.2 *Every vertex in* $\overline{U}$ *has capacity at least* 2.

Proof If any vertex v has capacity 1 then $x_v = 1$ (Step 1). Hence, all capacity 1 vertices belong to U. ■

Lemma 4.3 *Let* $e = (u, v)$ *and* $v \in helper(u)$. *Then* $y'_{ev} = 1$. *In other words, vertex* v *contributes* 1 *towards* h_u.

Proof Since $v \in helper(u)$, $y'_{ev} = \min\{1, k_v - \sum_{f \in E'(v) \setminus \{e\}} y'_{fv}\}$. To prove our claim, we must show that $k_v - \sum_{f \in E'(v) \setminus \{e\}} y'_{fv} \geq 1$.

$$\begin{aligned}
k_v - \sum_{f \in E'(v) \setminus \{e\}} y'_{fv} &= k_v - \sum_{f \in E'(v) \setminus \{e\}} y_{fv} \\
&\geq k_v - \sum_{f \in E'(v)} y_{fv} \\
&= k_v - \sum_{f \in E(v)} y_{fv} \\
&\geq k_v - k_v x_v && \text{(Using constraint (1))} \\
&\geq k_v - k_v/2 && \text{(Since } x_v < 1/2\text{)} \\
&= k_v/2 \\
&\geq 1. && \text{(Since } k_v \geq 2\text{)}
\end{aligned}$$

■

Lemma 4.4 *Each vertex* $u \in P$ *is charged at most* $\lceil h_u \rceil$ *by vertices in* I', *i.e.,* $|helper(u)| \leq \lceil h_u \rceil$.

Proof The claim follows from Lemma 4.3 and from the fact that h_u could be a fraction. ■

Remark: Observe that if $x_u = 1/2$, we are done since $E'(u) = \emptyset$. Hence, whenever we need to calculate the expected cost of a vertex $u \in U$, we can assume $0 \leq \epsilon_u < 1/2$.

Lemma 4.5 *Let* $u \in U$. *Let* Z_u *be the random variable that denotes the help received by vertex* u *in Step* 4 *of the algorithm, i.e.,* $Z_u = \sum_{e=(u,v):v \in I} y_{ev}/x_v$. *Then* $\mu_u = \mathbf{E}[Z_u] \geq 2h_u(1 - \epsilon_u)/(1 - 2\epsilon_u)$.

Proof Recall that $h_u = \sum_{e=(u,v)\in E'(u)}(y_{ev} - \epsilon_u)/(1-\epsilon_u)$ and $d_u = |E'(u)|$. By definition of expectation, we have

$$\begin{aligned}
\mu_u &= \sum_{e=(u,v)\in E'(u)} (y_{ev}/x_v)2x_v \\
&= 2 \sum_{e=(u,v)\in E'(u)} y_{ev} \qquad (2) \\
&= 2(1-\epsilon_u)h_u + 2d_u\epsilon_u \\
&= 2h_u - 2h_u\epsilon_u + 2d_u\epsilon_u \\
&= 2h_u + 2\epsilon_u(d_u - h_u) \qquad (3)
\end{aligned}$$

Since $\mu_u \le d_u$, we have

$$\begin{aligned}
d_u &\ge 2h_u + 2\epsilon_u(d_u - h_u) \\
&= h_u + h_u + 2\epsilon_u d_u - 2\epsilon_u h_u \\
d_u(1-2\epsilon_u) &\ge h_u + h_u(1-2\epsilon_u) \\
(d_u - h_u)(1-2\epsilon_u) &\ge h_u \\
d_u - h_u &\ge \frac{h_u}{1-2\epsilon_u} \qquad (4)
\end{aligned}$$

From (3) and (4), we have

$$\begin{aligned}
\mu_u &\ge 2h_u + 2\epsilon_u h_u/(1-2\epsilon_u) \\
&= \frac{2h_u(1-\epsilon_u)}{1-2\epsilon_u}
\end{aligned}$$

■

Notation: From now on, let $\exp(x)$ denote e^x.

Lemma 4.6 *If X_u is the random variable denoting the charge on a vertex $u \in U$ due to vertices in I', then $\mathbf{E}[X_u] \le \sum_{i=0}^{\lfloor h_u \rfloor} \left(exp(-\delta_i)/(1-\delta_i)^{(1-\delta_i)} \right)^{\mu_u}$, for some $\delta_i \in [\frac{1}{2(1-\epsilon)} + \frac{i(1-2\epsilon)}{2h_u(1-\epsilon)}, 1]$ and $\mu_u = \mathbf{E}[Z_u]$. When $\delta_i = 1$, we evaluate the summand in the limit as $\delta_i \to 1$: this limit is $exp(-\mu_u)$.*

Proof Note that X_u can have any integer between 0 and $\lceil h_u \rceil$. By definition of expectation, we have

$$\begin{aligned}
\mathbf{E}[X_u] &= \sum_{i=1}^{\lceil h_u \rceil} i \Pr[X_u = i] \\
&= \sum_{i=0}^{\lfloor h_u \rfloor} \Pr[X_u \ge i+1] \\
&\le \sum_{i=0}^{\lfloor h_u \rfloor} \Pr[Z_u \le \lceil h_u \rceil - (i+1)] \\
&\le \sum_{i=0}^{\lfloor h_u \rfloor} \Pr[Z_u \le h_u - i] \qquad (5)
\end{aligned}$$

Since Z_u is a sum of independent random variables each lying in $[0,1]$, we get using the Chernoff-Hoeffding bound that

$$\Pr\left[Z_u \leq \mu_u(1-\delta_i)\right] \leq \left(\exp(-\delta_i)/(1-\delta_i)^{(1-\delta_i)}\right)^{\mu_u}$$

The value of δ_i can be obtained as follows.

$$\begin{aligned} 1-\delta_i &= \frac{h_u - i}{\mu_u} && (6)\\ &\leq \frac{(h_u-i)(1-2\epsilon_u)}{2h_u(1-\epsilon_u)} && \text{(Using Lemma 4.5)}\\ \delta_i &\geq 1 - \frac{(h_u-i)(1-2\epsilon_u)}{2h_u(1-\epsilon_u)}\\ &= \frac{h_u + i(1-2\epsilon_u)}{2h_u(1-\epsilon_u)}\\ &= \frac{1}{2(1-\epsilon_u)} + \frac{i(1-2\epsilon_u)}{2h_u(1-\epsilon_u)} \end{aligned}$$

■

Lemma 4.7 *For $0 \leq \delta < 1$, the function $f(\delta) = 1/(1-\delta)^{(1-\delta)}$ attains a maximum value of exp$(1/e)$ at $\delta = 1 - 1/e$.*

Proof

$$\begin{aligned} f(\delta) &= 1/(1-\delta)^{(1-\delta)}\\ &= \exp(-(1-\delta)\ln(1-\delta)).\\ f'(\delta) &= \exp(-(1-\delta)\ln(1-\delta))((1-\delta)/(1-\delta) + \ln(1-\delta))\\ &= \exp(-(1-\delta)\ln(1-\delta))(1+\ln(1-\delta)) \end{aligned}$$

Solving $f'(\delta) = 0$, we get $\delta = 1 - 1/e$. Since $f'(1-2/e)$ is positive and $f'(1-1/(2e))$ is negative, we conclude that $f(1-1/e) = \exp(1/e)$ is the maximum. ■

Lemma 4.8 *For any vertex $u \in U$, if $h_u \geq 2$ then* $\mathbf{E}[X_u] \leq 1 - 2\epsilon_u$.

Proof From Lemma 4.6 and Lemma 4.7, we get

$$\mathbf{E}[X_u] \leq \sum_{i=0}^{\lfloor h_u \rfloor} (\exp(1/e - \delta_i))^{\mu_u}$$

From Lemma 4.6, we know that $\forall i \geq 0$, $\delta_i \geq 1/2$. hence, $1/e - \delta_i$ is always negative. Also, μ_u is always positive. Hence, the summand is maximized when μ_u and δ_i are minimized. Thus, we get

$$\mathbf{E}[X_u] \leq \sum_{i=0}^{\lfloor h_u \rfloor} \left(\exp\left(\frac{1}{e} - \frac{h_u + i(1-2\epsilon_u)}{2h_u(1-\epsilon_u)}\right)\right)^{\frac{2h_u(1-\epsilon_u)}{1-2\epsilon_u}}$$

$$
\begin{aligned}
&= \sum_{i=0}^{\lfloor h_u \rfloor} \exp\left(\frac{2h_u(1-\epsilon_u)}{e(1-2\epsilon_u)} - \frac{h_u}{1-2\epsilon_u} - i\right) \\
&= \sum_{i=0}^{\lfloor h_u \rfloor} \exp(p-i) \qquad \left(\text{where } p = \frac{2h_u(1-\epsilon_u)}{e(1-2\epsilon_u)} - \frac{h_u}{1-2\epsilon_u}\right) \\
&= \sum_{i=0}^{\lfloor h_u \rfloor} \frac{\exp(p)}{\exp(i)} \\
&= \frac{e}{e-1} \cdot \exp(p) \cdot (1 - \exp(-\lfloor h_u \rfloor - 1)) \\
&\leq \frac{e}{e-1} \cdot \exp(p) \cdot (1 - \exp(-h_u - 1)). \qquad (7)
\end{aligned}
$$

We will now show that $f(h_u) = \exp(p)(1 - \exp(-h_u - 1)$ is a decreasing function of h_u. $f'(h_u) = \exp(p)(\frac{2(1-\epsilon_u)}{e(1-2\epsilon_u)} - \frac{1}{1-2\epsilon_u}) - \exp(p - h_u - 1)(\frac{2(1-\epsilon_u)}{e(1-2\epsilon_u)} - \frac{1}{1-2\epsilon_u} - 1)$. The expression $\left(\frac{2(1-\epsilon_u)}{e(1-2\epsilon_u)} - \frac{1}{1-2\epsilon_u}\right)$ is negative since $2(1-\epsilon_u)/e < 1$. Since the first term dominates the second term $f'(h_u)$ is negative. Thus $f(h_u)$ is decreasing and is maximized when h_u is minimized. When $h_u = 2$,

$$p = \frac{4(1-\epsilon_u)}{e(1-2\epsilon_u)} - \frac{2}{1-2\epsilon_u} = \frac{2}{e} - \frac{K_1}{1-2\epsilon_u},$$

where K_1 is the positive constant $(2e-2)/e$. Thus, from (7), it is sufficient to show that

$$\forall \epsilon \in [0, 1/2),\ K_2 \cdot \exp(-K_1/(1-2\epsilon)) \leq 1 - 2\epsilon,$$

where K_2 is the constant $\frac{e^2+e+1}{e^2} \cdot \exp(2/e)$. Making the substitution $\psi = \frac{1}{1-2\epsilon}$ and taking the natural logarithm on both sides, it suffices to show:

$$\forall \psi \geq 1,\ -\ln\psi + K_1\psi - \ln K_2 \geq 0.$$

The inequality holds for $\psi = 1$. Also, for $\psi > 1$, the function $\psi \mapsto -\ln\psi + K_1\psi - \ln K_2$ has derivative $K_1 - 1/\psi$; since $K_1 = 2 - 2/e$ is greater than 1, the function increases for $\psi > 1$, and so we are done. ■

Lemma 4.9 *For any vertex $u \in U$, if $0 < h_u < 1$ then $\mathbf{E}[X_u] \leq 1 - 2\epsilon_u$.*

Proof Recall that $d_u = |E'(u)|$. Consider the case when $d_u = 1$. Let $e = (u, v) \in E'(u)$. Thus, $h_u \leq \epsilon_u \leq y_{ev} \leq x_v$. Thus, with a probability of $2x_v \geq 2\epsilon_u$, $v \in I$ and u receives the help h_u. Hence, the probability with which u participates in Step 5, i.e., $u \in P$, is $1 - 2\epsilon_u$. In that case, $|helper(u)| \leq 1$. Hence, $\mathbf{E}[X_u] \leq 1 - 2\epsilon_u$. Now consider when $d_u = 2$. Let $e_1 = (u, v)$ and $e_2 = (u, w)$ be the edges in $E'(u)$. Since $\mu_u = 2(y_{e_1v} + y_{e_2w}) \geq 4\epsilon_u$. From (3), we know that $\mu_u \geq 2h_u$. Hence, either $h_{e_1v} \geq h_u$ or $h_{e_2w} \geq h_u$. Without loss of generality, let $h_{e_1v} \geq h_u$. Since $x_v \geq \epsilon_u$, the probability of u receiving help of h_u in the randomized rounding step (Step 4) is at least $2\epsilon_u$. Hence, u participates in Step 5 (Alteration Step) of the algorithm with a probability of at most $2(1 - \epsilon_u)$. Thus, we get $\mathbf{E}[X_u] \leq 1 - 2\epsilon_u$. For the remainder of the lemma we assume that $d_u \geq 3$.

From inequality (5), we know that $\mathbf{E}[X_u] \leq \Pr[Z_u \leq h_u]$. Recall that Z_u is the random variable that represents the amount of help that u receives in Step 4 (Randomized Rounding) of the algorithm. Let $Z_u = \sum_{e=(u,v)\in E'(u)} Z_{ev}$, where Z_{ev} is the random variable that denotes the amount of help that v provides to u in Step 4 of the algorithm. Next suppose X is a random variable with mean μ and variance σ^2; suppose $a > 0$. Then, the well-known Chebyshev's inequality states that $\Pr[|X-\mu| \geq a]$ is at most σ^2/a^2. We will need stronger tail bounds than this, but only on X's deviation *below* its mean. The Chebyshev-Cantelli inequality shows that

$$\Pr[X \leq \mu - a] \leq \sigma^2/(\sigma^2 + a^2). \tag{8}$$

Define

$$y_u = \frac{\sum_{e=(u,v)\in E'(u)} y_{ev}}{d_u},$$

and note that

$$\epsilon_u \leq y_u \leq 1/2. \tag{9}$$

We will use (8) to bound $\Pr[Z_u \leq h_u]$. Thus, setting $\mu_u - a = h_u$ and using (2) we get

$$\begin{aligned} a &= \mu_u - h_u \\ &= 2 \sum_{e=(u,v)\in E'(u)} y_{ev} - \sum_{e=(u,v)\in E'(u)} (y_{ev} - \epsilon_u)/(1-\epsilon_u) \\ &= 2d_u y_u - \frac{d_u y_u - d_u \epsilon_u}{1-\epsilon_u} \\ &= \frac{d_u y_u - 2d_u\epsilon_u y_u + d_u \epsilon_u}{1-\epsilon_u} \\ &= d_u \left(2y_u - \frac{y_u - \epsilon_u}{1-\epsilon_u}\right) \end{aligned} \tag{10}$$

Let σ_u^2 and σ_{ev}^2 denote the variance of the random variables Z_u and Z_{ev} respectively. Since Z_u is the sum of the *independent* random variables Z_{ev}, we get

$$\begin{aligned} \sigma_u^2 &= \sum_{e=(u,v)\in E'(u)} \sigma_{ev}^2 \\ &= \sum_{e=(u,v)\in E'(u)} \mathbf{E}[Z_{ev}^2] - \mathbf{E}[Z_{ev}]^2 \\ &= \sum_{e=(u,v)\in E'(u)} \left(2x_v\left(\frac{y_{ev}^2}{x_v^2}\right) - 4y_{ev}^2\right) \\ &= \sum_{e=(u,v)\in E'(u)} \frac{2y_{ev}^2}{x_v} - 4y_{ev}^2 \end{aligned} \tag{11}$$

For a fixed a, the r.h.s. of (8) is maximized when σ^2 is maximized. We know that $\epsilon_u \leq y_{ev} \leq x_v < 1/2$. The R.H.S of (11) is maximized when x_v is minimized. Also, for a fixed value of $\sum_{e=u,v\in E'(u)} y_{ev}$, $\sum_{e=(u,v)\in E'(u)} y_{ev}^2$ is minimized when $y_{ev} = y_{e'v'} = y_u$, $\forall e = (u,v) \in E'(u)$ and $e' = (u,v') \in E'(u)$. Note that we are not changing the value of $\sum_{e=(u,v)\in E'(u)} y_{ev}$. Substituting $y_{ev} = y_u$ and $x_v = y_{ev} = y_u$ in the above inequality, we get

$$\begin{aligned} \sigma_u^2 &\leq \sum_{e=(u,v)\in E'(u)} 2y_u(1-2y_u) \\ &\leq 2d_u y_u(1-2y_u) \end{aligned}$$

Using (8), the value of a from (10), and the value of σ_u^2 from (12), we get

$$\begin{aligned}
\mathbf{E}[X_u] &\leq \Pr[Z_u \leq h_u] \\
&\leq \frac{\sigma_u^2}{\sigma_u^2 + a^2} \\
&\leq \frac{2d_u y_u(1-2y_u)}{2d_u y_u(1-2y_u) + d_u^2\left(2y_u - \frac{y_u - \epsilon_u}{1-\epsilon_u}\right)^2} \\
&= \frac{2y_u(1-2y_u)}{2y_u(1-2y_u) + d_u\left(2y_u - \frac{y_u - \epsilon_u}{1-\epsilon_u}\right)^2} \\
&\leq \frac{2y_u(1-2y_u)}{2y_u(1-2y_u) + 3\left(2y_u - \frac{y_u - \epsilon_u}{1-\epsilon_u}\right)^2} \qquad (12)
\end{aligned}$$

We will analyze the cost by considering the following two cases.

Case I: $\epsilon > 3y_u/4$.

We would like to upper-bound the value of $\frac{y_u - \epsilon_u}{1-\epsilon_u}$ in (12). For $\epsilon_u \leq y_u < 4\epsilon_u/3$, we will calculate a value $c \in [0,1]$ such that $\frac{y_u - \epsilon_u}{1-\epsilon_u} \leq c y_u$.

$$\begin{aligned}
c &\geq \frac{y_u - \epsilon_u}{y_u(1-\epsilon_u)} \\
&= \frac{1}{1-\epsilon_u} - \frac{\epsilon_u}{y_u(1-\epsilon_u)} \\
&\geq \frac{1}{1-\epsilon_u} - \frac{\epsilon_u}{(4\epsilon_u/3)(1-\epsilon_u)} \\
&= \frac{1}{1-\epsilon_u} - \frac{3}{4(1-\epsilon_u)} \\
&= \frac{1}{4(1-\epsilon_u)} \\
&\geq \frac{1}{4(1-(1/2))} \qquad \text{(Since } \epsilon_u \leq 1/2\text{)} \\
&= 1/2.
\end{aligned}$$

Thus, substituting the value of $(y_u - \epsilon_u)/(1-\epsilon_u)$ as $y_u/2$ in (12), we get

$$\begin{aligned}
\mathbf{E}[X_u] &\leq \frac{2y_u(1-2y_u)}{2y_u(1-2y_u) + 3\left(2y_u - y_u/2\right)^2} \\
&= \frac{2y_u(1-2y_u)}{2y_u - 4y_u^2 + (27y_u^2/4)} \\
&\leq \frac{2y_u(1-2y_u)}{2y_u} \\
&= 1 - 2y_u.
\end{aligned}$$

Case II: $\epsilon \leq 3y_u/4$.

We want to show that $\mathbf{E}[X_u] \leq 1 - 2\epsilon_u$. From (12), we conclude that it is sufficient to show that

$$\frac{2y_u(1-2y_u)}{2y_u(1-2y_u) + 3\left(2y_u - \frac{y_u - \epsilon_u}{1-\epsilon_u}\right)^2} \leq 1 - 2\epsilon_u$$

$$2y_u(1-2y_u) - 2y_u(1-2y_u)(1-2\epsilon_u) \quad \leq \quad 3\left(2y_u - \frac{y_u-\epsilon_u}{1-\epsilon_u}\right)^2 (1-2\epsilon_u) \tag{13}$$

We will consider the L.H.S and R.H.S of (13) separately.

$$\begin{aligned}
\text{L.H.S} &= 2y_u(1-2y_u) - 2y_u(1-2y_u)(1-2\epsilon_u) \\
&= 2y_u(1-2y_u)(2\epsilon_u) \\
&= 4\epsilon_u y_u(1-2y_u) \\
&\leq 3y_u^2(1-2y_u) \qquad\qquad \text{(Since } \epsilon_u \leq 3y_u/4) \qquad (14)
\end{aligned}$$

$$\begin{aligned}
\text{R.H.S} &= 3\left(2y_u - \frac{y_u-\epsilon_u}{1-\epsilon_u}\right)^2 (1-2\epsilon_u) \\
&= 3\left(\frac{2y_u - 2y_u\epsilon_u - y_u + \epsilon_u}{1-\epsilon_u}\right)^2 (1-2\epsilon_u) \\
&= 3\left(\frac{y_u - 2y_u\epsilon_u + \epsilon_u}{1-\epsilon_u}\right)^2 (1-2\epsilon_u) \\
&= 3\left(\frac{y_u}{1-\epsilon_u} + \frac{\epsilon_u(1-2y_u)}{1-\epsilon_u}\right)^2 (1-2\epsilon_u) \\
&\geq 3y_u^2(1-2\epsilon_u) \qquad\qquad \text{(Since } y_u < 1/2) \qquad (15)
\end{aligned}$$

From (14) and (15) we conclude that L.H.S$\leq$ R.H.S and $\mathbf{E}[X_u] \leq 1-2\epsilon_u$. ■

Lemma 4.10 *For any vertex $u \in U$, if $1 \leq h_u < 2$ then $\mathbf{E}[X_u] \leq 1-2\epsilon_u$.*

Proof We will use the notation d_u, μ_u, y_u, σ_u^2 etc. as in the proof of Lemma 4.9. As in that proof, we have

$$\sigma_u^2 \leq 2d_u y_u(1-2y_u).$$

Recall that $h_u = d_u(y_u-\epsilon_u)/(1-\epsilon_u)$. Thus, by Chebyshev-Cantelli,

$$\begin{aligned}
\mathbf{E}[X_u] &\leq \Pr\left[Z_u \leq h_u - 1\right] + \Pr\left[Z_u \leq h_u\right] \\
&\leq \frac{2d_u y_u(1-2y_u)}{2d_u y_u(1-2y_u) + (\mu_u - h_u + 1)^2} + \frac{2d_u y_u(1-2y_u)}{2d_u y_u(1-2y_u) + (\mu_u - h_u)^2} \\
&\leq \frac{2d_u y_u(1-2y_u)}{2d_u y_u(1-2y_u) + (2d_u y_u - d_u(y_u-\epsilon_u)/(1-\epsilon_u) + 1)^2} + \\
&\quad \frac{2d_u y_u(1-2y_u)}{2d_u y_u(1-2y_u) + (2d_u y_u - d_u(y_u-\epsilon_u)/(1-\epsilon_u))^2} \\
&= \frac{2d_u y_u(1-2y_u)}{2d_u y_u(1-2y_u) + (2d_u y_u - \frac{d_u(y_u-\epsilon_u)}{1-\epsilon_u} + 1)^2} + \frac{2y_u(1-2y_u)}{2y_u(1-2y_u) + d_u(2y_u - \frac{y_u-\epsilon_u}{1-\epsilon_u})^2}. \qquad (16)
\end{aligned}$$

Now fix ϵ_u and y_u arbitrarily (subject to the constraints $0 \leq \epsilon_u < y_u \leq 1/2$), and consider an adversary who wishes to maximize (16) subject to the constraint that d_u is a real number for which

$$d_u(y_u-\epsilon_u)/(1-\epsilon_u) \geq 1. \tag{17}$$

It is then sufficient to show that the maximum value (achievable by the adversary) is at most $1-2\epsilon_u$; we will do so now. It is clear that the second term in (16) is maximized when $d_u(y_u-\epsilon_u)/(1-\epsilon_u) = 1$. We now show that this is also true for the first term in (16). Maximizing this term is equivalent to minimizing its reciprocal, which is equivalent to minimizing

$$\frac{(2d_u y_u - d_u(y_u-\epsilon_u)/(1-\epsilon_u)+1)^2}{d_u}.$$

The derivative of this term w.r.t. d_u is

$$(2y_u-(y_u-\epsilon_u)/(1-\epsilon_u))^2 - 1/d_u^2 \geq y_u^2 - 1/d_u^2 \geq 0,$$

since $d_u y_u \geq 1$ by (17). Thus, (16) is maximized when $d_u(y_u-\epsilon_u)/(1-\epsilon_u) = 1$; making the substitution $z = 2d_u y_u$, we make some observations:

since $z = 2y_u(1-\epsilon_u)/(y_u-\epsilon_u)$ where $0 \leq \epsilon_u < y_u$, we have $z \geq 2$; also, $\epsilon_u = \frac{y_u(z-2)}{z-2y_u}$.

So, to show that (16) is at most $1-2\epsilon_u$, we need to show

$$\begin{aligned}
\frac{2d_u y_u(1-2y_u)}{2d_u y_u(1-2y_u)+(2d_u y_u)^2} + \frac{2d_u y_u(1-2y_u)}{2d_u y_u(1-2y_u)+(2d_u y_u-1)^2} &\leq 1-2\epsilon_u; \text{ i.e.,} \\
\frac{1-2y_u}{1-2y_u+z} + \frac{z(1-2y_u)}{z(1-2y_u)+(z-1)^2} &\leq 1-\frac{2y_u(z-2)}{z-2y_u}; \text{ i.e.,} \\
\frac{z}{1-2y_u+z} - \frac{z(1-2y_u)}{z(1-2y_u)+(z-1)^2} &\geq \frac{2y_u(z-2)}{z-2y_u}. \qquad (18)
\end{aligned}$$

Substitute $p = 1-2y_u$, and note that $p \in [0,1]$. Simplifying (18), we want to show that

$$z\cdot(z+p-1)\cdot((z-1)^2-p^2) \geq (1-p)\cdot(z-2)\cdot(z+p)\cdot((z-1)^2+pz).$$

Since $z \geq 2$ and $0 \leq p \leq 1$, all the factors in this last inequality are non-negative; so, it suffices to show that

$$\begin{aligned}
z &\geq (1-p)\cdot(z+p), \text{ and} \\
(z+p-1)\cdot((z-1)^2-p^2) &\geq (z-2)\cdot((z-1)^2+pz).
\end{aligned}$$

The first inequality reduces to $zp \geq p(1-p)$, which is true since $z \geq 2 > 1-p$. The second inequality reduces to $-p^3-p^2(z-1)+p+(z-1)^2 \geq 0$. For a fixed p, the derivative of the l.h.s. (w.r.t. z) is easily to seen to be non-negative for $z \geq 2$. Thus, it suffices to check that $-p^3-p^2(z-1)+p+(z-1)^2 \geq 0$ is non-negative when $z=2$, which follows from the fact that $p \in [0,1]$. ■

Theorem 4.11 *Let Cost be the random variable that represents the cost of our vertex cover, C. Then* $\mathbf{E}[Cost] \leq 2OPT$.

Proof Let $Cost(u)$ be the random variable that is denotes the cost incurred by a vertex $u \in V$. For $u \in U$, $Cost(u) = 1+X_u$, where recall that X_u is the random variable that represents the charge on U due to vertices chosen in Step 5 (Alteration Step) of the algorithm. For $u \in \overline{U}$, $Cost(u) = 1$ if it is chosen in Step 4 (Randomized Rounding) of the algorithm and 0 otherwise.

$$
\begin{aligned}
Cost &= \sum_{u \in V} Cost(u) \\
&= \sum_{u \in U} Cost(u) + \sum_{u \in \overline{U}} Cost(u) \\
&= \sum_{u \in U} (1 + X_u) + \sum_{u \in \overline{U}} Cost(u) \\
\mathbf{E}[Cost] &= \sum_{u \in U} (1 + \mathbf{E}[X_u]) + \sum_{u \in \overline{U}} \mathbf{E}[Cost(u)] \\
&\leq \sum_{u \in U} (1 + 1 - 2\epsilon_u) + \sum_{u \in \overline{U}} 1 \cdot \Pr[u \in I] \\
&= \sum_{u \in U} 2(1 - \epsilon_u) + \sum_{u \in \overline{U}} 2x_u \\
&\leq 2 \cdot OPT.
\end{aligned}
$$

■

Acknowledgments. We thank Seffi Naor for helpful discussions on the problem considered.

References

[1] R. Bar-Yehuda and S. Even. *A Local-Ratio Theorem for Approximating The Weighted Vertex Cover Problem.* In Annals of Discrete Mathematics, 25:27-45, 1985.

[2] J. Bar-Ilan, G. Kortsarz and D. Peleg, Generalized submodular cover problems and applications, *Theoretical Computer Science*, 250, pages 179-200, 2001.

[3] Chv'atal V. A Greedy Heuristic for the Set Covering Problem, *Mathematics of Operations Research*, vol. 4, No 3, 1979, pp. 233-235.

[4] R. D. Carr, L. K. Fleischer, V. J. Leung and C. A. Phillips. *Strengthening Integrality Gaps For Capacitated Network Design and Covering Problems.* In Proc. of the 11th ACM-SIAM Symposium on Discrete Algorithms, 106-115, 2000.

[5] J. Chuzhoy and J. Naor. *Covering Problems with Hard Capacities.* In Proc. of the Forty-Third IEEE Symposium on Foundations of Computer Science, 481-489, 2002.

[6] G. Dobson. *Worst Case Analysis of Greedy Heuristics For Integer Programming with Non-Negative Data.* Mathematics of Operations Research, 7(4):515-531, 1980.

[7] R. Gandhi, S. Khuller, S. Parthasarathy and A. Srinivasan. *Dependent Rounding in Bipartite Graphs.* In Proc. of the Forty-Third IEEE Symposium on Foundations of Computer Science, pages 323-332, 2002.

[8] S. Guha, R. Hassin, S. Khuller and E. Or. *Capacitated Vertex Covering with Applications.* In Proc. ACM-SIAM Symposium on Discrete Algorithms, pages 858-865, 2002.

[9] D. S. Johnson, "Approximation algorithms for combinatorial problems," J. Computer and System Sciences, 9 (1974) 256-278.

[10] E. Halperin. Improved approximation algorithms for the vertex cover problem in graphs and hypergraphs. In *Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms, San Francisco, California*, pages 329–337, 2000.

[11] D. S. Hochbaum. *Approximation Algorithms for the Set Covering and Vertex Cover Problems* SIAM Journal on Computing, 11:555-556, 1982.

[12] D. S. Hochbaum. *Heuristics for fixed cost median problem* Mathematical Programming, 22:148-162, 1982.

[13] D. S. Hochbaum (editor). *Approximation Algorithms for NP-hard Problems.* PWS Publishing Company, 1996.

[14] S. G. Kolliopoulos and N. E. Young. *Tight Approximation Results for General Covering Integer Programs.* In Proc. of the Forty-Second Annual Symposium on Foundations of Computer Science, 522-528, 2001.

[15] L. Lovasz, "On the ratio of optimal integral and fractional covers," Discrete Math., 13 (1975) 383-390.

[16] M. Pál, É. Tardos and T. Wexler. *Facility Location with Nonuniform Hard Capacities.* In Proc. Forty-Second Annual Symposium on Foundations of Computer Science, 329-338, 2001.

[17] V. Vazirani. *Approximation Algorithms.* Springer-Verlag, 2001.

[18] L. A. Wolsey. An analysis of the greedy algorithm for the submodular set covering problem. Combinatorica, 2:385–393, 1982.

[19] N. E. Young. K-medians, facility location, and the Chernoff-Wald bound *ACM-SIAM Symposium on Discrete Algorithms* 86-95.