An Improved Approximation Algorithm For Vertex Cover with Hard Capacities*

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

Proposed running head: Vertex Cover with Hard Capacities

*A preliminary version of this work appeared in the *Proc. International Colloquium on Automata, Languages, and Programming*, pages 164–175, 2003.

[†]Department of Computer Science, Rutgers University, Camden, NJ 08102. Part of this work was done while the author was a student at the University of Maryland, College Park and was supported by NSF Award CCR-9820965. E-mail: rajivg@camden.rutgers.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 Awards CCF-0430650 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. This material is based upon work supported in part by the National Science Foundation under Grants 0208005 and CNS-0426683. E-mail: srin@cs.umd.edu.

Abstract

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 can 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, approximation algorithms with an approximation factor of 2 were developed with the assumption that an arbitrary number of copies of a vertex may be chosen in the cover. If we are allowed to pick at most a fixed number of copies of each vertex, the approximation algorithm becomes much more complex. Chuzhoy and Naor (*FOCS, 2002*) have shown that the weighted version of this problem is at least as hard as set cover; in addition, they 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.

Key Words and Phrases: Approximation algorithms, capacitated covering, set cover, vertex cover, linear programming, randomized rounding.

1 Introduction

Covering problems such as set cover and facility location are fundamental in combinatorial optimization. Recent years have witnessed much interest in the *capacitated* versions of such covering problems, modeling, e.g., facilities that can serve a bounded number of customers, facilities that cannot be replicated an unbounded number of times, etc. In particular, capacitated versions of the vertex cover problem have received attention recently. We improve on previous results and present a 2-approximation algorithm for this problem; this cannot be improved further unless we can solve the standard (uncapacitated) vertex cover problem to within a factor better than 2.

Background. Vertex cover is a special case of the set-cover problem; recall that set cover requires the selection of a minimum number (or minimum cost) collection of subsets that cover a given universe. The set-cover problem with hard capacities generalizes the set-cover problem in that sets have capacity bounds on the number of elements that they can cover. In a seminal paper, Johnson gave the first (greedy) logarithmic ratio approximation for the unweighted uncapacitated set cover problem [10]. This was generalized by Chvátal [3] to the weighted uncapacitated case, and further generalized by Dobson [6] to approximating to within a logarithmic ratio the integer linear program min $c \cdot x$ subject to $Ax \geq b$, with all the entries in A non-negative integers. A much more general result is given by Wolsey [20], giving a logarithmic ratio approximation algorithm for submodular covering problems. Vertex cover and set cover with hard capacities are both examples of submodular covering problems. Hence [20] gave the first nontrivial (logarithmic) approximation for the capacitated versions of these problems. Research has also been conducted on the multi-set multi-cover problem. In this problem, the input sets are 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 integer program (IP): $\min\{w^T x | Ax \geq d, 0 \leq x \leq b, x \in Z\}$. The natural linear programming (LP) relaxation of this problem has an unbounded integrality gap. Dobson [6] gave a greedy algorithm achieving a guarantee of $H(\max_{1 \leq j \leq n} A_{ij})$; here, $H(t)$ is the Harmonic function $\sum_{i=1}^{t} 1/i$. 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 [15] have presented an $O(\log n)$–approximation algorithm for this problem.

A problem closely related to set cover with hard capacities is facility location with hard capacities. Here, we are given a set of facilities F and a set of clients C. There is a cost function c which defines the cost of assigning a client to a facility. Each facility $j \in F$ has a cost f_j, a bound b_j denoting the number of available copies of j and capacity k_j 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 total cost of open facilities and the cost of assigning the clients to them. A logarithmic greedy approximation problem for the uncapacitated case appears in [13] and for the capacitated case and some generalizations in [2]. Slightly improved (still logarithmic) bounds for the uncapacitated case are presented in [21] using randomized methods. For the case of *metric* facility location with hard capacities, Pál, Tardos and Wexler [17] gave a $(9 + \epsilon)$-approximation algorithm using local search.

All of the above results involve substantially more work for the capacitated case, as compared to the uncapacitated cases. This is also true in our context, as we describe next.

Our Problem and Results. The capacitated vertex cover problem can be defined as follows. Let $G = (V, E)$ be an undirected graph with vertex set V and edge set E. We are also given three non-negative quantities for each vertex v: a *weight* w_v, *capacity* k_v, and "number of allowed copies" b_v. We assume that k_v and b_v are integers. 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 $k_v x_v$. (In words, we can choose at most b_v copies of v; each such copy can cover at most k_v edges incident on 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 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 the standard vertex cover problem, by picking a node v in the cover, we can cover all edges incident to v; in the problem at hand, choosing one copy of v lets us cover at most k_v edges incident on v.

Motivated by an application in glycobiology, Guha *et al.* [9] studied the version of the problem in which b_v is unbounded. They obtain an approximation algorithm, with an approximation factor of 2, using a primal-dual approach. They also gave a 4-approximate solution using LP-rounding. Gandhi *et al.* [8] gave a 2-approximate solution using LP-rounding for the same problem. The problem becomes significantly harder when the values b_v are bounded. For arbitrary weights on the vertices, the work of Chuzhoy and Naor [5] shows the surprising result that the problem is at least as hard to approximate as the set cover problem; thus, an approximation guarantee of $(1-\epsilon)\ln n$ for any positive constant ϵ, would imply that $NP \subseteq DTIME[n^{\log\log n}]$ [7] (see also [18]). For the unweighted case (i.e., where $w_v = 1$ for all v), an elegant 3-approximation algorithm for this problem is presented in [5]. This algorithm uses randomized rounding of an LP relaxation followed by an alteration step. The algorithm and its analysis are quite subtle: indeed, the combination of bounded capacities and copies seems to be highly nontrivial to deal with.

In this paper, we modify the algorithm of Chuzhoy and Naor in two crucial ways to obtain a 2-approximate solution for the unweighted case. We add a pre-processing step in which we fix the number of copies of certain capacity-1 vertices. After fixing the number of copies of these vertices we solve the relaxation of the integer linear program. We also modify their alteration step in an important way that helps bound the cost of the alteration step in a better way. The best-known approximation algorithms for the standard (uncapacitated) vertex cover problem achieve an approximation ratio of $(2-o(1))$ for arbitrary graphs [1, 11, 12]; see [14] for a nice overview. It is an outstanding open question if the problem can be approximated to within $(2-\Omega(1))$. Since any improvement to our 2–approximation would immediately yield such an improvement for standard vertex cover, it appears challenging to improve our approximation to $(2-\Omega(1))$.

The rest of this paper is organized as follows. Sections 2 and 3 describe a natural IP formulation of the problem and our algorithm, respectively. Our algorithm is then analyzed in Section 4, and concluding remarks are made in Section 5.

2 IP Formulation and Relaxation

A natural IP formulation of the problem is as follows, as in [9]. In this formulation, $y_{ev} = 1$ if and only if the edge e is covered by vertex v. Clearly, the values of x in a feasible solution correspond to a capacitated cover. While we do not need the constraint "$\forall v \in e \in E,\ x_v \geq y_{ev}$" 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.

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

In the LP relaxation, we let the y_{ev} lie in $[0,1]$, and let each x_v be in the range $[0, b_v]$. We make a couple of observations regarding the IP formulation and this relaxation.

First, suppose we have the above IP formulation, and that we wish to check if there is a feasible *integral* solution. This can be done efficiently by applying a standard flow procedure. 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 each vertex in A_2 represents a vertex in V. An edge (e, v) is in F iff in G, the edge e is incident to vertex v. 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 $k_v b_v$. Now, there is a feasible solution to our problem iff the maximum flow value from the source to the sink is $|E|$.

Second, suppose we have a feasible solution (x', y') to the LP relaxation where x' is integral and y' is real; this can be converted to an integral solution (x, y) of no higher cost easily as follows. Construct a flow network just as in the previous paragraph, with the difference that the capacity of an edge going from a node representing $v \in V$ to the sink, is $k_v x'_v$. A maximum flow computation will give us the desired integral solution since there is always an integral flow in a network with integer capacities, of value the same as a fractional flow.

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 decide the number of copies of capacity-1 vertices to be included in our solution. 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 an integral 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 (as shown in Section 2). We intersperse the steps of the algorithm with a few clearly marked remarks, to give the reader a sense of how we are proceeding.

1. Pre-Processing. In this step, we try and diminish the values b_v for capacity-1 vertices v, as much as possible. Let b'_v denote the number of available copies of a vertex $v \in V$ at the end of this step. Initially, $b'_v = b_v$. For a vertex v which is not a capacity-1 vertex, b'_v does not change. For a capacity-1 vertex v, b'_v may change during the course of this step; this is done as follows. Consider the current n-dimensional vector $(b'_v : v \in V)$.

> Find some v (if any) so that $k_v = 1$ and reducing b'_v by 1 maintains feasibility (the feasibility-checking can be done as described in Section 2). If such a vertex v exists, then we set $b'_v \leftarrow b'_v - 1$ and repeat.

Finally, if $b'_v = 0$, then k_v is reset to 0. (Thus, for the rest of the discussion, any reference to a capacity-1 vertex would mean a capacity-1 vertex with a non-zero b'-value.)

Remark: We note that in the end of the preprocessing step, the optimal value of the resulting instance has the same value as the original instance - we prove this in Lemma 4.3. Thus, at the end of this step, we are left with a set of non-negative integer values $b'_v \leq b_v$ for all v, such that:

(P1) $b'_v = b_v$ if $k_v \geq 2$;

(P2) the IP formulation with b_v replaced by b'_v for all v, has a feasible solution; and

(P3) the IP formulation becomes infeasible if we decrease b'_v for any one capacity-1 vertex v, while keeping all other b' values the same.

2. LP Solution. Solve the LP relaxation *with the additional constraint "$x_v = b'_v$ for each capacity-1 vertex v"*. We have from (P2) that this problem is feasible. Let (x, y) denote the solution of this LP relaxation.

To facilitate the discussion of the remainder of the algorithm, let us introduce some notation.

- $U \doteq \{u \mid x_u \geq 1/2\}$.
- $\overline{U} \doteq V \setminus U$.
- $E' \doteq \{(u,v) \mid u \in U,\ v \in \overline{U}, (u,v) \in E\}$.
- Recall that $E(u)$ denotes the set of edges incident on u.
- $\forall u \in U,\ \epsilon_u \doteq \frac{\lceil x_u \rceil - x_u}{\lceil x_u \rceil}$; note that $0 \leq \epsilon_u \leq 1/2$ since $x_u \geq 1/2$.

Note that there are no edges within $\overline{U}$.

3. Partial Cover. We remark that from this point on, our goal is to construct a feasible solution (x', y') to the LP relaxation, where x' is integral.

Let $u \in U$. First, we "round it up": we set $x'_u \doteq \lceil x_u \rceil$. Next, for any edge $e = (u,v) \in E \setminus E'$, set $y'_{eu} = y_{eu}$ and $y'_{ev} = y_{ev}$. Also, for each edge $e = (u,v) \in E'$, with $u \in U$, define:

$$y'_{eu} \quad = \quad \min(1, y_{eu}\lceil x_u \rceil / x_u) = \min(1, \frac{(1 - y_{ev})\lceil x_u \rceil}{\lceil x_u \rceil (1 - \epsilon_u)}) = \min(1, \frac{1 - y_{ev}}{1 - \epsilon_u});$$

$$h_{ev} = \begin{cases} 0 & \text{if } y'_{eu} = 1 \\ 1 - y'_{eu} = \frac{y_{ev} - \epsilon_u}{1 - \epsilon_u} & \text{otherwise} \end{cases}$$

Define, for all $u \in U$: $E'(u) \doteq \{e = (u,v) | e \in E' \wedge h_{ev} > 0\}$ and $d_u \doteq |E'(u)|$. We also define $E''(u) \doteq \{e = (u,v) | e \in E' \wedge h_{ev} = 0\}$. Similarly, for $v \in \overline{U}$, $E'(v) \doteq \{e = (u,v) | e \in E' \wedge h_{ev} > 0\}$ and $d_v \doteq |E'(v)|$. For $u \in U$, define $h_u = \sum_{e=(u,v) \in E'(u)} h_{ev}$.

Remark: Since we have rounded u up from x_u to $x'_u = \lceil x_u \rceil$, the contribution of u toward covering edge $e = (u,v) \in E'(u)$ is y'_{eu}. (See also Fig. 1.) To cover all the edges in $E'(u)$ fractionally, we are going to need an additional coverage of $h_u = \sum_{e=(u,v) \in E'(u)} h_{ev}$. Note that for the remaining edges, they are fully (fractionally) covered by nodes in U. In the following steps we will get the necessary additional coverage from vertices in $\overline{U}$. (If we consider a solution where we only pick nodes in U, then each node u has an excess of h_u of fractional demand assigned to it.)

4. Randomized Rounding. Round each vertex $v \in \overline{U}$ to 1 (i.e., set $x'_v = 1$) *independently* with probability $2x_v$. Let I be the set of vertices that are rounded to 1 in this step. For each edge $e = (u, v) \in E'(v)$ such that $v \in I$, define $y'_{ev} = y_{ev}/x_v$. Also reset $y'_{eu} = 1 - y'_{ev}$.

Remark: y'_{ev} is the contribution of v towards covering e. By constraint (1), $\sum_{e \in E'(v)} y_{ev}/x_v = \sum_{e \in E'(v)} y'_{ev} \le k_v$. In fact for all nodes chosen in I we have $y'_{ev} \ge y_{ev} \ge h_{ev}$.

5. Alteration. Let $P \subseteq U$ be the vertices that still need some help from vertices in $\overline{U}$, i.e., $P = \{u \in U : \sum_{e=(u,v) \in E'(u), v \in I} y'_{ev} < h_u\}$.

Remark: In this step, we will choose a set of vertices $I' \subseteq \overline{U} \setminus I$, such that

$$\forall u \in P, \qquad \sum_{e=(u,v) \in E'(u), v \in I} y'_{ev} + \sum_{e=(u,v), v \in I'} y'_{ev} \ge h_u,$$

where for each vertex $v \in I'$, y'_{ev} is set according to step (c) below. For each vertex $u \in P$, we define a set of vertices helpers(u). Each vertex in helpers(u) contributes towards h_u. Each vertex in I' belongs to exactly one such set helpers(u).

Initially, $I' \leftarrow \emptyset$ and helpers$(u) \leftarrow \emptyset, \forall u \in P$. Perform the following four steps until P is empty:

(a) Pick an arbitrary vertex $u \in P$.

(b) Consider any edge (u, v) such that $v \in \overline{U} \setminus (I \cup I')$. Do the following:

- $x'_v \leftarrow 1$;
- helpers$(u) \leftarrow$ helpers$(u) \cup \{v\}$; and
- $I' \leftarrow I' \cup \{v\}$.

Now let $P'_v = \{w \in P : w \ne u, e' = (w, v) \in E'\}$.

(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 $y'_{ev} = 1$ and $y'_{eu} = 0$, where $e = (u, v)$. (We will prove that this does not violate the capacity of v in Lemma 4.5.)

(d) For each vertex $w \in (\{u\} \cup P'_v)$, if $\sum_{e=(w,a), a \in I \cup I'} y'_{ea} \ge h_w$, then remove w from P.

Now that P is empty, we have a feasible solution (x', y') in which x' is integral and y' may be fractional. For each edge $e = (u, v) \in E'$ such that $v \notin I \cup I'$, set $y'_{ev} = 0$ and $y'_{eu} = 1$. In addition, set $x'_v = 0$ for $v \in \overline{U} \setminus (I \cup I')$.

6. Integral Solution. Convert (x', y') to an integral solution of no higher cost, as shown in Section 2.

This completes the description of the algorithm.

4 Analysis

In Step 5 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 $v \in$ helpers(u). We will charge u the cost of adding v to our solution. Note that in the LP solution the cost of vertex u is $x_u = \lceil x_u \rceil (1 - \epsilon_u)$. In our solution, vertex $u \in U$ pays for itself and for the vertices in helpers(u).

Our primary goal. Our primary goal will be to show that for any $u \in U$, the total expected charge on u due to vertices in helpers(u) is at most $\lceil x_u \rceil (1 - 2\epsilon_u)$.

Suppose we can achieve this goal. Thus, the total expected cost of vertex $u \in U$ is at most $\lceil x_u \rceil (1 - 2\epsilon_u) + \lceil x_u \rceil = 2\lceil x_u \rceil (1 - \epsilon_u) = 2x_u$. Also, the total expected size of I is $\sum_{v \in \overline{U}} 2x_v$. Thus we will obtain a 2-approximation in expectation, by using the linearity of expectation.

Theorem 4.1 *Assume that the expected charge on u due to vertices in helpers(u) is at most $\lceil x_u \rceil (1-2\epsilon_u)$. Let $Cost$ be the random variable that represents the cost of our vertex cover, C. Then $\mathbf{E}[Cost] \leq 2 \cdot OPT$.*

Proof We can define $Cost$ as $\sum_{u \in U} \lceil x_u \rceil + |I| + |I'|$. Thus $\mathbf{E}[Cost] = \sum_{u \in U} \lceil x_u \rceil + \mathbf{E}[|I|] + \mathbf{E}[|I'|]$. By the claim mentioned earlier, we can charge the cost of I' to nodes in U so that the expected cost of each node $u \in U$ is $\lceil x_u \rceil (1 - 2\epsilon_u)$. We thus obtain Thus $\mathbf{E}[Cost] = \sum_{u \in U} (\lceil x_u \rceil + \lceil x_u \rceil (1 - 2\epsilon_u)) + \sum_{u \in \overline{U}} 2x_u \leq \sum_{u \in U} 2x_u + \sum_{u \in \overline{U}} 2x_u \leq 2OPT$. ■

In addition, observe that in Step 5(b) a vertex $v \in \overline{U} \setminus (I \cup I')$ always exists. This is because u is still in P, and $\sum_{e=(u,a), a \in I \cup I'} y'_{ea} < h_u$. By definition of h_u we can see that there are nodes in $v \in \overline{U} \setminus (I \cup I')$ that can be chosen.

Theorem 4.2 *The solution (x', y') obtained by the algorithm has the property that x' is integral, and this is a feasible solution for the relaxation of the IP in Section 2.*

Proof First recall that for $u \in U$: $E'(u) \doteq \{e = (u,v) | e \in E' \wedge h_{ev} > 0\}$ and $d_u \doteq |E'(u)|$. In addition $E''(u) \doteq \{e = (u,v) | e \in E' \wedge h_{ev} = 0\}$. Similarly, for $v \in \overline{U}$ $E'(v) \doteq \{e = (u,v) | e \in E' \wedge h_{ev} > 0\}$ and $d_v \doteq |E'(v)|$.

We argue that all edges are covered fractionally. First note that $\overline{U}$ is an independent set in the graph. The edges in $E \setminus E'$ are all covered fractionally, as their end vertices are both in U and we round x_u to $\lceil x_u \rceil$. For the remaining edges $e = (u,v) \in E'$, each edge is provided a coverage of y'_{eu} once we round x_u to $\lceil x_u \rceil$. We could modify this later, but ensure that $y'_{eu} + y'_{ev} = 1$. In Step 5 of the algorithm, we make sure that all vertices in P have neighbors chosen (vertices in I') to provide coverage at least h_u (total deficiency at u).

We next argue that we do not violate the capacity of any vertex. In other words, each vertex covers only $k_v \cdot x'_v$ edges. For a vertex $v \in I$ it is easy to see that $x'_v = 1$ and the total fractional load is $\sum_{e \in E'(v)} y'_{ev} = \sum_{e \in E'(v)} \frac{y_{ev}}{x_v} \leq k_v$. This follows since $\sum_{e \in E(v)} y_{ev} \leq k_v x_v$. For a vertex $v \in I'$, $x'_v = 1$ and Lemma 4.5 ensures that the capacity is not violated.

For a vertex $u \in U$ (after Step 3), we have the following (by the feasible LP solution)

$$\sum_{e \in E(u) \setminus (E'(u) \cup E''(u))} y_{eu} + \sum_{e \in E'(u) \cup E''(u)} y_{eu} \leq k_u x_u$$

Multiplying both sides by $\frac{\lceil x_u \rceil}{x_u} \geq 1$.

$$\sum_{e \in E(u) \setminus (E'(u) \cup E''(u))} y'_{eu} + \sum_{e \in E'(u) \cup E''(u)} y'_{eu} \leq k_u \lceil x_u \rceil.$$

$$\sum_{e \in E(u) \setminus (E'(u) \cup E''(u))} y'_{eu} + \sum_{e \in E'(u)} y'_{eu} + |E''(u)| \leq k_u \lceil x_u \rceil.$$

Note that the capacity of this vertex at the end of the algorithm is $k_u\lceil x_u \rceil$. At this stage, *if* we set $x'_v = 0$ for all vertices $v \in \overline{U}$, then the total demand assigned to u will be as follows.

$$\sum_{e \in E(u) \setminus (E'(u) \cup E''(u))} y'_{eu} + |E'(u)| + |E''(u)| \le \sum_{e \in E(u) \setminus (E'(u) \cup E''(u))} y'_{eu} + \sum_{e \in E'(u)} y'_{eu} + \sum_{e \in E'(u)} (1 - y'_{eu}) + |E''(u)|$$

$$\le k_u \lceil x_u \rceil + h_u.$$

Steps 4 and 5 guarantee that h_u amount of coverage will be provided by the neighbors of u in $\overline{U}$. Thus we are able to get a fractional y' that satisfies the LP constraints (with an integral x'). Note that each step we when add nodes to $I \cup I'$ these newly nodes take away some of the demand assigned to u. When the demand assigned to u reduces by at least h_u, we get a valid cover where all the edges are fractionally covered. ■

Step 6 ensures that we find an integral covering from the fractional covering produced in Step 5.

4.1 Preliminaries

We first show that our preprocessing step is justifiable. Lemma 4.3 shows that the optimal integral solution value remains unchanged after the preprocessing step; in particular, the cost of the LP solution for the preprocessed graph is a lower bound on the cost of of the original instance, which is what we need.

Lemma 4.3 *Let G_o be the original graph instance. Let G_n be the new graph instance that results after the pre-processing step (Step 1). Let $OPT(G_o)$ and $OPT(G_n)$ represent the optimal solutions in G_o and G_n respectively. We claim that the two optimal solutions have the same cost.*

Proof Let $R = \{v : b'_v < b_v\}$. Observe that each vertex in R is a capacity-1 vertex. For a vertex $v \in R$ and any solution S, let N_v^S denote the number of copies of v used by solution S. Let $OPT(G_o)$ be an optimal solution S to G_o in which the following potential function Φ is minimized:

$$\Phi(S) \doteq \sum_{v:\ N^S(v) > b'_v} (N_v^S - b'_v).$$

(Note that if $N^S(v) > b'_v$, then $v \in R$.) First suppose this minimum value of Φ is zero; thus, there exists an optimal solution to the original instance in which for all v, the number of copies of v used is at most b'_v. However, we know from (P3) – see the remark in Step 1 of the algorithm – that in any such solution, we must have $x_v = b'_v$ for each capacity-1 vertex v. Thus, the extra constraint imposed in Step 2 of the algorithm does not change the set of feasible solutions of the IP, and the proof if completed.

So, suppose the minimum value of Φ is positive. For convenience, we simply let N_v denote the number of copies of vertex v used by $OPT(G_o)$. Thus, we are now in the case where the set $R' = \{v \in R : N_v > b'_v\}$ is nonempty. We now present a proof by contradiction that R' must be empty.

Construct a directed graph H having the same vertex set as G_o. Include an arc (a, b) in H iff edge (a, b) in G_o is covered by a in $OPT(G_o)$ and by b in $OPT(G_n)$. Note that every element of R' has outdegree strictly larger than its indegree in H. We now construct a directed path Q in H as follows. Initialize a directed graph H' to H. If there is a simple cycle in H', delete all its edges; repeat this until no cycles are left in H'. Note that it is still true that every element of R' has outdegree strictly larger than its indegree in H'. Thus, H' is now a directed acyclic graph with at least one edge. Let v be an arbitrary element of

R' and let Q be a maximal path in H' starting from v. Let w be the last vertex in the path. Now consider this simple path Q in H, and note that w has its indegree strictly more than its out-degree, in H.

Consider a new solution to the original instance which is the same as $OPT(G_o)$, except that: (i) the number of copies of vertex v is reduced by 1, and (ii) a new copy of w is added to the solution iff the total number of copies of w in $OPT(G_o)$ is at most $b'_w - 1$. In the new solution, let the edges of Q have the same assignment as in $OPT(G_n)$; the assignment of edges to all other vertices remain the same as in $OPT(G_o)$. We will now show that this new assignment does not violate the capacity constraint (constraint (1) in LP) of any vertex. The only vertices that are affected are the vertices in Q. Since one edge is "moved away" from v and since v has capacity 1, it is feasible to remove one copy of v; this is where we use the fact that v has capacity 1. What about w? First let us consider the case in which no new copy of w is added to the solution. Since w has its indegree strictly more than its out-degree in H, w covers at least one more edge in $OPT(G_n)$ than it covers in $OPT(G_o)$. Since $OPT(G_o)$ and $OPT(G_n)$ both use b'_w copies of w, the total capacity of w ($k_w b'_w$) is the same in $OPT(G_o)$ and in $OPT(G_n)$. Thus, w covers at most $k_w b'_w - 1$ 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. The cost of this solution is the same as $OPT(G_o)$ and it decreases Φ by one, contradicting the minimality of $OPT(G_o)$ w.r.t. Φ. Now consider the case when a new copy of w is added to the solution. Again, the cost of the new solution is the same as that of $OPT(G_o)$ and Φ is lessened again: note that although we add a copy of w, w still does not contribute to Φ since its new number of copies is at most b'_w.

Thus we see that R must be empty, concluding the proof. ■

Lemma 4.4 *Every vertex in $\overline{U}$ has capacity at least* 2.

Proof If a vertex v has capacity 1, then x_v is a positive integer (Step 1). Hence, all capacity-1 vertices belong to U. ■

Lemma 4.5 *Let $e = (u, v)$ and $v \in helpers(u)$. Vertex v can contribute* 1 *towards h_u without violating its capacity.*

Proof Since $v \in \text{helpers}(u)$, we have that $y'_{ev} = 1$. To prove our claim, we must show that $\sum_{f=(w,v)\wedge f\neq e} y'_{fv} + 1 \leq k_v$. The L.H.S. evaluates to

$$\sum_{f\in E(v)\setminus\{e\}} y_{fv} + 1 \leq \sum_{f\in E(v)} y_{fv} + 1.$$

Using constraint (1), we get that the L.H.S. is at most $k_v x_v + 1 \leq k_v/2 + 1 \leq k_v$. This is true since $k_v \geq 2$. ■

In particular, we deduce:

Lemma 4.6 *Each vertex $u \in P$ is charged at most $\lceil h_u \rceil$ by vertices in I', i.e., $|helpers(u)| \leq \lceil h_u \rceil$.*

Remark: Recall our primary goal from the beginning of this section. If $x_u = 1/2$, then the goal is trivially achieved since $E'(u) = \emptyset$. Hence, whenever we need to calculate the expected cost of a vertex $u \in U$, we assume from now on that $0 \le \epsilon_u < 1/2$.

We next define a couple of key random variables.

Notation: Let $u \in U$. We let Z_u be the random variable denoting the help received by vertex u in Step 4 of the algorithm, i.e., $Z_u = \sum_{e=(u,v)\in E'(u): v \in I} y_{ev}/x_v$. Also, X_u is the random variable denoting the total charge on u due to vertices in I'.

Lemma 4.7 *Let $u \in U$. Then, $\mu_u \doteq \boldsymbol{E}[Z_u] \ge 2h_u(1-\epsilon_u)/(1-2\epsilon_u)$.*

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

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

Since $d_u \ge \mu_u$, we have $d_u \ge 2h_u + 2\epsilon_u(d_u - h_u)$. This gives us $d_u - h_u \ge h_u/(1-2\epsilon_u)$. Combining this inequality with (3), we get $\mu_u \ge 2h_u + 2\epsilon_u h_u/(1-2\epsilon_u) = 2h_u(1-\epsilon_u)/(1-2\epsilon_u)$. ■

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

Lemma 4.8 *Consider any $u \in U$. Let $\mu_u = \boldsymbol{E}[Z_u]$. Then,*

$$\boldsymbol{E}[X_u] \le \sum_{i=0}^{\lfloor h_u \rfloor} \left(exp(-\delta_i)/(1-\delta_i)^{(1-\delta_i)} \right)^{\mu_u},$$

where each δ_i lies in the interval $[\frac{1}{2(1-\epsilon_u)} + \frac{i(1-2\epsilon_u)}{2h_u(1-\epsilon_u)}, 1]$. When $\delta_i = 1$, we evaluate the summand in the limit as $\delta_i \to 1$ from above: this limit is $exp(-\mu_u)$.

Proof Note that X_u lies in the set $\{0, 1, \ldots, \lceil h_u \rceil\}$. By definition of expectation, we have

$$\mathbf{E}[X_u] = \sum_{i=1}^{\lceil h_u \rceil} i \cdot \Pr[X_u = i] \le \sum_{i=0}^{\lfloor h_u \rfloor} \Pr[X_u \ge i+1] \le \sum_{i=0}^{\lfloor h_u \rfloor} \Pr[Z_u \le \lceil h_u \rceil - (i+1)].$$

Thus we get

$$\mathbf{E}[X_u] \le \sum_{i=0}^{\lfloor h_u \rfloor} \Pr[Z_u \le h_u - i] \qquad (4)$$

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

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

when $\delta_i = 1$, the R.H.S. is interpreted in the limit as δ_i tends to one from above; i.e., in this case, the R.H.S. is taken to be $\exp(-\mu_u)$.

The value of δ_i is given by

$$1 - \delta_i = \frac{h_u - i}{\mu_u}. \tag{5}$$

Combining (5) with Lemma 4.7, we get $\delta_i \geq \frac{1}{2(1-\epsilon_u)} + \frac{i(1-2\epsilon_u)}{2h_u(1-\epsilon_u)}$. ■

The following has an elementary proof, which is omitted:

Lemma 4.9 *For $0 \leq \delta < 1$, the function $\delta \mapsto 1/(1-\delta)^{(1-\delta)}$ attains a maximum value of exp$(1/e)$ at $\delta = 1 - 1/e$.*

4.2 The analysis of three cases

Our primary goal is to show that for any $u \in U$, $\mathbf{E}[X_u] \leq (1 - 2\epsilon_u) \cdot \lceil x_u \rceil$. We now show a stronger version of this: that $\mathbf{E}[X_u] \leq 1 - 2\epsilon_u$. We do this via three lemmas, which handle different ranges of the value h_u.

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

Proof From Lemma 4.8 and Lemma 4.9, we get $\mathbf{E}[X_u] \leq \sum_{i=0}^{\lfloor h_u \rfloor} (\exp(1/e - \delta_i))^{\mu_u}$. From Lemma 4.8, 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

$$\begin{aligned}
\mathbf{E}[X_u] &\leq \sum_{i=0}^{\lfloor h_u \rfloor} \left(\exp\left(\frac{1}{e} - \frac{h_u + i(1-2\epsilon_u)}{2h_u(1-\epsilon_u)}\right)\right)^{\frac{2h_u(1-\epsilon_u)}{1-2\epsilon_u}} \\
&= \sum_{i=0}^{\lfloor h_u \rfloor} \exp(p - i) \qquad \left(\text{where } p = \frac{2h_u(1-\epsilon_u)}{e(1-2\epsilon_u)} - \frac{h_u}{1-2\epsilon_u}\right) \\
&= \sum_{i=0}^{\lfloor h_u \rfloor} \frac{\exp(p)}{\exp(i)} \\
&= \frac{e}{e-1} \cdot \exp(p) \cdot (1 - \exp(-\lfloor h_u \rfloor - 1)) \\
&\leq \frac{e}{e-1} \cdot \exp(p) \cdot (1 - \exp(-h_u - 1)). \qquad (6)
\end{aligned}$$

We will now show that $f(h_u) = \exp(p) \cdot (1 - \exp(-h_u - 1))$ is a decreasing function of h_u. Note that

$$f'(h_u) = \exp(p) \cdot (\frac{2(1-\epsilon_u)}{e(1-2\epsilon_u)} - \frac{1}{1-2\epsilon_u}) - \exp(p - h_u - 1) \cdot (\frac{2(1-\epsilon_u)}{e(1-2\epsilon_u)} - \frac{1}{1-2\epsilon_u} - 1).$$

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 (6), it is sufficient to show that

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

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:

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

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. ■

The next two lemmas handle the case where $h_u < 2$. The Chernoff-Hoeffding bound-based approach does not seem strong enough in this case, and we resort to another approach in the proofs of these lemmas.

Lemma 4.11 *For any vertex $u \in U$, if $0 < h_u < 1$ then $\mathbf{E}[X_u] \leq 1 - 2\epsilon_u$.*

Proof We start with a useful observation. Suppose $e = (u, v) \in E'(u)$. Then,

$$x_v \geq y_{ev} = 1 - y_{eu} > 1 - x_u/\lceil x_u \rceil = \epsilon_u. \tag{7}$$

The inequality follows because $y_{eu} < x_u/\lceil x_u \rceil$; otherwise $y_{eu}\lceil x_u \rceil / x_u \geq 1$ implying that $h_{ev} = 0$, which is not possible as we are assuming that $h_{ev} > 0$ since the edge is in $E'(u)$.

Recall that $d_u = |E'(u)|$. Consider first the case $d_u = 1$. 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 at most $1 - 2\epsilon_u$. In that case, $|\text{helpers}(u)| \leq 1$. Hence, $\mathbf{E}[X_u] \leq 1 - 2\epsilon_u$. Next consider the case $d_u = 2$. Let $e_1 = (u, v)$ and $e_2 = (u, w)$ be the edges in $E'(u)$. From (3), we know that $\mu_u \geq 2h_u$. Since the expected help received from the two neighbors v and w is at least $2h_u$, the help received from one of the two neighbors v or w must be at least h_u if its chosen in I. Assume that the help received from v is at least 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 $1 - 2\epsilon_u$; if it does participate, then $X_u = 1$ with probability 1 by Lemma 4.6, since $h_u \leq 1$. Thus, we get $\mathbf{E}[X_u] \leq 1 - 2\epsilon_u$.

For the remainder of the proof, we assume that $d_u \geq 3$. Since $h_u < 1$, we know from (4) that $\mathbf{E}[X_u] \leq \Pr[Z_u \leq h_u]$. 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

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

note that

$$0 \leq \epsilon_u < y_u < 1/2, \tag{9}$$

since $\epsilon_u < y_{ev} \leq x_v < 1/2$ by (7), and since $|E'(u)| = d_u$. We will use (8) to bound $\Pr[Z_u \leq h_u]$. Thus, setting $\mu_u - a = h_u$ and using (2) we get

$$a = \mu_u - h_u = 2 \sum_{e=(u,v)\in E'(u)} y_{ev} - \sum_{e=(u,v)\in E'(u)} (y_{ev} - \epsilon_u)/(1 - \epsilon_u) = 2d_u y_u - (d_u y_u - d_u \epsilon_u)/(1 - \epsilon_u).$$

This gives us

$$a = d_u \left(2y_u - \frac{y_u - \epsilon_u}{1-\epsilon_u}\right). \tag{10}$$

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

$$\sigma_u^2 = \sum_{e=(u,v)\in E'(u)} \sigma_{ev}^2 = \sum_{e=(u,v)\in E'(u)} (\mathbf{E}[Z_{ev}^2] - \mathbf{E}[Z_{ev}]^2).$$

This gives us

$$\sigma_u^2 = \sum_{e=(u,v)\in E'(u)} (\frac{2y_{ev}^2}{x_v} - 4y_{ev}^2). \tag{11}$$

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 (11), we get

$$\sigma_u^2 \leq \sum_{e=(u,v)\in E'(u)} 2y_u(1-2y_u) = 2d_u y_u(1-2y_u). \tag{12}$$

Using (8), (10), and (12), we get

$$\begin{aligned}
\mathbf{E}[X_u] &\leq \Pr[Z_u \leq h_u] \\
&\leq \sigma_u^2/(\sigma_u^2 + a^2) \\
&\leq \frac{2d_u y_u(1-2y_u)}{2d_u y_u(1-2y_u) + d_u^2\left(2y_u - (y_u-\epsilon_u)/(1-\epsilon_u)\right)^2} \\
&\leq \frac{2y_u(1-2y_u)}{2y_u(1-2y_u) + 3\left(2y_u - \frac{y_u-\epsilon_u}{1-\epsilon_u}\right)^2},
\end{aligned} \tag{13}$$

since $d_u \geq 3$.

We will now consider two cases:

Case I: $\epsilon_u > 3y_u/4$.
We would like to upper-bound the value of $(y_u - \epsilon_u)/(1-\epsilon_u)$ in (13). We have

$$\begin{aligned}
(y_u - \epsilon_u)/(y_u(1-\epsilon_u)) &= 1/(1-\epsilon_u) - \epsilon_u/(y_u(1-\epsilon_u)) \\
&\leq 1/(1-\epsilon_u) - \epsilon_u/((4\epsilon_u/3)(1-\epsilon_u)) \\
&= 1/(4(1-\epsilon_u)) \\
&\leq 1/2.
\end{aligned}$$

Thus, substituting the value of $(y_u - \epsilon_u)/(1-\epsilon_u)$ as $y_u/2$ in (13), we get

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

Case II: $\epsilon_u \le 3y_u/4$.

We want to show that $\mathbf{E}[X_u] \le 1 - 2\epsilon_u$. Thus, it is sufficient to show that the R.H.S. of (13) is at most $1 - 2\epsilon_u$; i.e., it is sufficient to show that

$$2y_u(1-2y_u) - 2y_u(1-2y_u)(1-2\epsilon_u) \le 3\left(2y_u - \frac{y_u - \epsilon_u}{1-\epsilon_u}\right)^2 (1-2\epsilon_u). \tag{14}$$

We will consider the L.H.S. and R.H.S. of (14) separately. L.H.S. $= 2y_u(1-2y_u) - 2y_u(1-2y_u)(1-2\epsilon_u) = 2y_u(1-2y_u)(2\epsilon_u) = 4\epsilon_u y_u(1-2y_u)$. Since $\epsilon_u \le 3y_u/4$, we get

$$\text{L.H.S.} \le 3y_u^2(1-2y_u). \tag{15}$$

The R.H.S. evaluates to $3\left(y_u/(1-\epsilon_u) + \epsilon_u(1-2y_u)/(1-\epsilon_u)\right)^2 (1-2\epsilon_u)$. Since $y_u < 1/2$, we get

$$\text{R.H.S} \ge 3y_u^2(1-2\epsilon_u). \tag{16}$$

From (15) and (16) we conclude that L.H.S. $\le$ R.H.S. and hence that $\mathbf{E}[X_u] \le 1 - 2\epsilon_u$. ■

Lemma 4.12 *For any vertex $u \in U$, if $1 \le h_u < 2$ then $\mathbf{E}[X_u] \le 1 - 2\epsilon_u$.*

Proof We will use the notation d_u, μ_u, y_u, σ_u^2 etc. as in the proof of Lemma 4.11. As in that proof, we have $\sigma_u^2 \le 2d_u y_u(1-2y_u)$. Recall that $h_u = d_u(y_u - \epsilon_u)/(1-\epsilon_u)$; also, we have $\mu_u = 2d_u y_u$ by (2). So, by Chebyshev-Cantelli,

$$\begin{aligned} \mathbf{E}[X_u] &\le \Pr[Z_u \le h_u - 1] + \Pr[Z_u \le h_u] \\ &\le \frac{2d_u y_u(1-2y_u)}{2d_u y_u(1-2y_u) + (\mu_u - h_u + 1)^2} + \frac{2d_u y_u(1-2y_u)}{2d_u y_u(1-2y_u) + (\mu_u - h_u)^2} \\ &= \frac{2d_u y_u(1-2y_u)}{2d_u y_u(1-2y_u) + (2d_u y_u - \frac{d_u(y_u-\epsilon_u)}{1-\epsilon_u} + 1)^2} + \frac{2y_u(1-2y_u)}{2y_u(1-2y_u) + d_u(2y_u - \frac{y_u-\epsilon_u}{1-\epsilon_u})^2} \end{aligned} \tag{17}$$

Now fix ϵ_u and y_u arbitrarily (subject to the constraints $0 \le \epsilon_u < y_u \le 1/2$), and consider an adversary who wishes to maximize (17) subject to the constraint that d_u is a real number for which $d_u(y_u - \epsilon_u)/(1-\epsilon_u) \ge 1$. 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.

We now show that (17) is maximized when $d_u(y_u - \epsilon_u)/(1-\epsilon_u) = 1$. It is clear that the second term in (17) 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 (17). 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

$$(2y_u - (y_u - \epsilon_u)/(1-\epsilon_u))^2 - 1/d_u^2 \ge y_u^2 - 1/d_u^2;$$

the fact that this is non-negative follows from the fact that $d_u y_u \ge 1$ (which in turn holds, since $d_u(y_u - \epsilon_u)/(1-\epsilon_u) \ge 1$.) So, to show that (17) is at most $1 - 2\epsilon_u$, we need to show that

$$\frac{2d_u y_u(1-2y_u)}{2d_u y_u(1-2y_u) + (2d_u y_u)^2} + \frac{2d_u y_u(1-2y_u)}{2d_u y_u(1-2y_u) + (2d_u y_u - 1)^2} \le 1 - 2\epsilon_u. \tag{18}$$

Making the substitution $z = 2d_u y_u$, we make some observations. Since $z = 2y_u(1-\epsilon_u)/(y_u - \epsilon_u)$ where $0 \le \epsilon_u < y_u$, we have $z \ge 2$; also, $\epsilon_u = \frac{y_u(z-2)}{z-2y_u}$. So, the required bound (18) becomes

$$\frac{1-2y_u}{1-2y_u+z} + \frac{z(1-2y_u)}{z(1-2y_u)+(z-1)^2} \le 1 - \frac{2y_u(z-2)}{z-2y_u};$$

i.e., we want to show that

$$\frac{z}{1-2y_u+z} - \frac{z(1-2y_u)}{z(1-2y_u)+(z-1)^2} \ge \frac{2y_u(z-2)}{z-2y_u}. \tag{19}$$

Substitute $p = 1-2y_u$, and note that $p \in [0,1]$. Simplifying (19), we want to show that

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

Since $z \ge 2$ and $0 \le p \le 1$, all the factors in this last inequality are non-negative; so, it suffices to show that $z \ge (1-p)\cdot(z+p)$, and $(z+p-1)\cdot((z-1)^2-p^2) \ge (z-2)\cdot((z-1)^2+pz)$. The first inequality reduces to $zp \ge p(1-p)$, which is true since $z \ge 2 > 1-p$. The second inequality reduces to $-p^3 - p^2(z-1) + p + (z-1)^2 \ge 0$. For a fixed p, the derivative of the L.H.S. (w.r.t. z) is easily seen to be non-negative for $z \ge 2$. Therefore, it suffices to check that $-p^3 - p^2(z-1) + p + (z-1)^2 \ge 0$ is non-negative when $z = 2$, which follows from the fact that $p \in [0,1]$. ■

5 Conclusion

We have presented what appears to be the best-possible approximation algorithm for the unweighted capacitated vertex-cover problem with hard constraints. It would be interesting to see if there is a combinatorial approximation algorithm achieving the same approximation ratio.

Acknowledgments. We thank Refael Hassin and Seffi Naor for helpful discussions. We thank the referees for their very useful comments.

References

[1] R. Bar-Yehuda and S. Even. *A Local-Ratio Theorem for Approximating The Weighted Vertex Cover Problem.* 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, 250:179-200, 2001.

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

[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, pages 106-115, 2000.

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

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

[7] U. Feige. *A Threshold of* $\ln n$ *for Approximating Set Cover.* In Proceedings of the 28th Annual ACM Symposium on Theory of Computing, pages 182-189, 1996.

[8] 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.

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

[10] D. S. Johnson. *Approximation algorithms for combinatorial problems.* J. Computer and System Sciences, 9, pages 256-278, 1974.

[11] 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.

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

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

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

[15] 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, pages 522-528, 2001.

[16] L. Lovász. *On the ratio of optimal integral and fractional covers.* Discrete Math., 13, pages 383-390, 1975.

[17] 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.

[18] R. Raz and S. Safra. *A Sub-Constant Error-Probability Low-Degree Test, and a Sub-Constant Error-Probability PCP Characterization of NP.* In Proceedings of the 29th Annual ACM Symposium on Theory of Computing, pages 475-484, 1997.

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

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

[21] N. E. Young. *K-medians, facility location, and the Chernoff-Wald bound.* ACM-SIAM Symposium on Discrete Algorithms, pages 86-95, 2000.

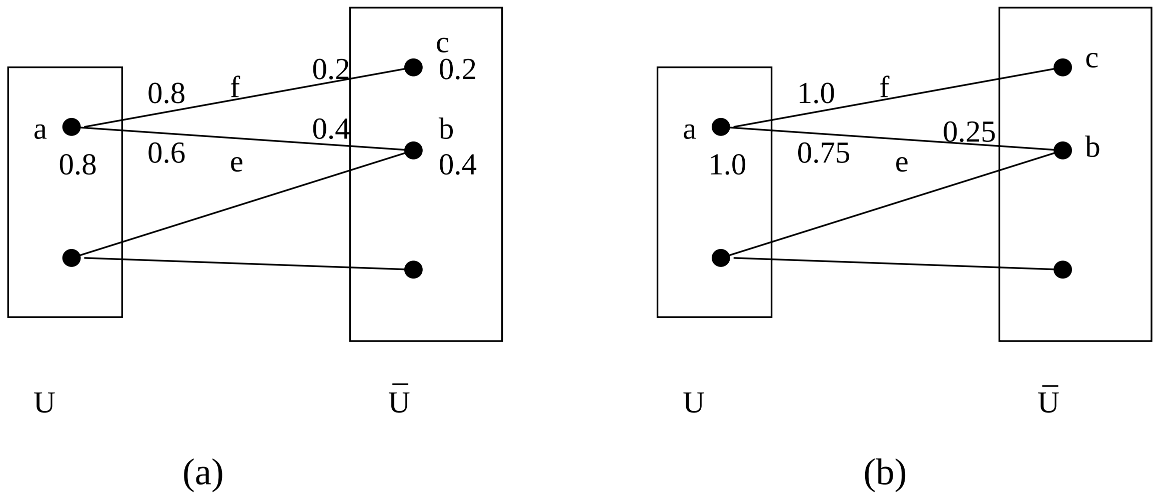

Figure 1: In (a) we have $x_a = 0.8$ and $x_b = 0.4$. In (b) we set $x'_a = 1$. After Step 3, $y'_{ea} = 0.75$. Note that $h_{eb} = 0.25$. Also notice that $y'_{fa} = 1.0$ and this edge is not in $E'(a)$. If $b \in I$ then we define $y'_{eb} = 1$ and redefine $y'_{ea} = 0$ (Step 4).