Covering Problems with Hard Capacities

Julia Chuzhoy Joseph (Seffi) Naor

Computer Science Department
Technion, Haifa 32000, Israel
E-mail: {*cjulia,naor*}*@cs.technion.ac.il.*

Abstract

We consider the classical vertex cover and set cover problems with the addition of hard *capacity constraints. This means that a set (vertex) can only cover a limited number of its elements (adjacent edges) and the number of available copies of each set (vertex) is bounded. This is a natural generalization of the classical problems that also captures resource limitations in practical scenarios.*

We obtain the following results. For the unweighted vertex cover problem with hard capacities we give a 3*-approximation algorithm which is based on randomized rounding with alterations. We prove that the weighted version is at least as hard as the set cover problem. This is an interesting separation between the approximability of weighted and unweighted versions of a "natural" graph problem. A logarithmic approximation factor for both the set cover and the weighted vertex cover problem with hard capacities follows from the work of Wolsey [23] on submodular set cover. We provide in this paper a simple and intuitive proof for this bound.*

1. Introduction

The *set cover* problem is defined as follows. Let $E = \{1, \dots, n\}$ be a ground set and let $\mathcal{S}$ be a collection of sets defined over E. Each $S \in \mathcal{S}$ has a non-negative *cost* $w(S)$ associated with it. A *cover* is a collection of sets such that their union is E. The goal is to find a cover of minimum cost. The set cover problem is a classic NP-hard problem that was studied extensively in the literature, and the best approximation factor achievable for it is $\Theta(\log n)$ [7, 9, 15, 17].

We consider in this paper the set cover problem with *capacity* constraints, or the *capacitated set cover* problem. Here, each set $S \in \mathcal{S}$ has a *capacity* $k(S)$ and a *multiplicity* $m(S)$ associated with it, meaning that at most $m(S)$ copies of set S can be used and each copy can cover at most $k(S)$ elements of S. A cover $\mathcal{C}$ is a multi-set of input sets that can cover all the elements, while $\mathcal{C}$ contains at most $m(S)$ copies of each $S \in \mathcal{S}$, and each copy covers at most $k(S)$ elements. We assume that the capacity constraints are *hard*, i.e., the number of copies and the capacity of a set cannot be exceeded. The capacitated set cover problem is a natural generalization of a basic and well-studied problem that captures practical scenarios where resource limitations are present.

A special case of the capacitated set cover problem that we consider is the *capacitated vertex cover* problem, defined as follows. An undirected graph $G = (V, E)$ is given and each vertex $v \in V$ is associated with a *cost* $w(v)$, a *capacity* $k(v)$, and a *multiplicity* $m(v)$. The goal is to find a minimum cost multi-set U of vertices that cover all the edges, such that for each vertex $v \in V$, at most $m(v)$ copies appear in U, and each copy covers at most $k(v)$ edges adjacent to v. The capacitated vertex cover problem generalizes the well known vertex cover problem, probably one of the most studied problems (see [14] for an overview), for which the best currently known approximation factor is $2 - \frac{\log \log |V|}{2 \log |V|}$ [3, 13].

The capacitated vertex cover problem was first introduced by Guha, Hassin, Khuller and Or [12]. They considered the version of the problem with *soft* capacities, a special case where the number of available copies of each vertex is unbounded. A straightforward rounding of a linear programming relaxation of the problem gives a 4-approximate solution. Guha *et al.* [12] show a 2-approximation primal-dual algorithm and they also give a 3-approximation for the case where each edge $e \in E$ has an (unsplittable) demand $d(e)$. (Gandhi *et al.* [10] provide further results on the capacitated vertex cover problem with soft capacities.) Guha *et al.* [12] motivate the study of the capacitated vertex cover problem by an application in glycobiology. The problem emerged in the redesign of known drugs involving glycoproteins and can be represented as an instance of the capacitated vertex cover problem.

A closely related problem is *facility location* with hard capacities. In this problem, the input consists of a set of facilities and a set of clients. For each facility and each client, there is a distance that defines the cost of assigning the client to the facility. Each facility f has a cost w_f, a capacity k_f, and a number of available copies m_f. Each client i has a demand d_i. The goal is to open facilities and to assign all the clients to them. The cost of the solution is the total cost of the open facilities plus the assignment costs of the clients. Each copy of facility f can serve total demand of at most k_f, and at most m_f such copies can be opened. The capacitated set cover problem is a special case of facility location with hard capacities, where all the distances are either 0 or ∞ (note that this distance function is not a metric). Bar-Ilan *et al.* [2] give an $O(\log n + \log M)$-approximation for the facility location with hard capacities, where M is the maximal input parameter.

Prior Work There is extensive research on the set cover problem and the reader is referred to the surveys in [11, 6, 1, 19, 14]. Feige [9] proved that it is impossible to obtain a better than $(1 + o(1)) \ln n$-approximation for set cover, unless NP has slightly super-polynomial time algorithms. A greedy heuristic gives an O($\log n$)-approximation [7, 17] for the set cover problem.

Wolsey [23] considered the *submodular set cover* problem. Let f be a real valued function defined over all subsets of a finite set N. Function f is called *non-decreasing* if $f(S) \leq f(T)$ for all $S \subseteq T \subseteq N$, and *submodular* if $f(S)+f(T) \geq f(S\cap T)+f(S\cup T)$ for all $S, T \subseteq N$. The input to the submodular set cover problem is a set cover instance together with a non-decreasing submodular function f defined over all collections of the input sets (i.e., here, $N = \mathcal{S}$). The goal is to find a minimum cost cover $\mathcal{C}$ such that $f(\mathcal{C}) = f(\mathcal{S})$. In the case of the capacitated set cover problem, for any multiset $\mathcal{A}$ of input sets, define $f(\mathcal{A})$ to be the maximum number of elements that $\mathcal{A}$ can cover (given the capacity constraints). It is not hard to see that f is a non-decreasing submodular function. Wolsey [23] showed using dual fitting that the approximation factor of a greedy heuristic for the submodular set cover problem is $O(\log f_{\max})$, where $f_{\max} = \max_{S\in\mathcal{S}} f(\{S\})$.

Metric facility location is a well studied field. Many heuristics, as well as approximation algorithms with bounded performance guarantees, were developed [5, 18, 20, 22]. For the metric facility location problem with hard capacities, Pál, Tardos and Wexler [21] recently gave a $(9 + \epsilon)$-approximation using local search.

1.1. Our Contribution

The first result we present is a 3-approximation for the unweighted capacitated vertex cover problem. Our algorithm uses randomized rounding with *alterations*. The first rounding step in our algorithm applies randomized rounding where the probabilities are derived from a solution to a linear programming relaxation of the problem. However, the rounding may not yield a feasible cover and therefore we need to add more vertices to the cover. This is done in the alteration step. Our analysis uses a sophisticated charging scheme to bound the number of vertices that are added to the cover in this step. We also prove that the more general version where edges have unsplitable demands is not approximable in the presence of hard capacities. Contrast this with the 3-approximation algorithm of Guha *et al.* [12] for this case (with soft capacities).

We consider the weighted capacitated vertex cover and prove that it is set-cover hard. This means that the best approximation factor that can be achieved for this problem is $\Omega(\log n)$. Our hardness proof holds even for the case of $\{0, 1\}$ weights and unit multiplicity. Interestingly, we are not aware of any other "natural" graph problem where there is a logarithmic separation between the approximability of the weighted and unweighted versions. (However, there are several examples of problems where the unweighted version is polynomially solvable while the weighted version is NP-hard.) We note that Gandhi *et al.* [10] obtained a 2-approximation algorithm for the weighted capacitated vertex cover problem where the capacity of each vertex can be exceeded by at most a factor of two.

We proceed to consider the capacitated set cover problem. As already noted, it follows from Wolsey's work [23] that a natural greedy heuristic achieves an approximation factor of $O(\log n)$ for this problem. We note that the integrality gap of the natural linear programming relaxation of the problem is unbounded, similar to the case of facility location with hard capacities [21]. Indeed, Wolsey used a different linear programming formulation (see Section 6 for a description of the linear program). We consider the same greedy heuristic as Wolsey and provide a direct combinatorial proof of the approximation factor of this heuristic. We believe that our proof is simple and intuitive. We note that the main obstacle in applying the "standard" (set cover) charging scheme in the presence of hard capacities is that it is not clear how to "charge" the sets in the optimal solution for the sets in the solution computed by the greedy algorithm. Since there are hard capacities, the assignment of elements to sets in the cover is dynamic, and, moreover, elements may be covered and uncovered several times during the iterations of the algorithm.

The rest of the paper is organized as follows. In Section 3, we show a 3-approximation algorithm for unweighted capacitated vertex cover. In section 4.1 we show that the weighted capacitated vertex cover problem is at least as hard as the set cover problem, even in the case where $m(v) = 1$ for all $v \in V$. In Section 5 we provide a description of the greedy algorithm for the set cover problem with hard capacities, and give a simple proof that the algorithm achieves an $O(\log n)$-approximation, implying an $O(\log |V|)$-approximation for the weighted capacitated vertex cover problem. In Section 6 we discuss extensions of the algorithm to more general covering problems, such as submodular set cover and multi-set multi-cover.

2. Preliminaries

Consider an instance of the set cover problem with hard capacities. Let $\mathcal{P}$ be a multiset of sets from $\mathcal{S}$, where each $S \in \mathcal{S}$ appears in $\mathcal{P}$ at most $m(S)$ times. Then $C \subseteq \mathcal{P} \times E$ defines a partial cover of the elements by the sets from $\mathcal{P}$: for each $S \in \mathcal{P}, e \in S$, such that if $(S, e) \in C$, we say that e is covered by S in C. The cover C is called *feasible*, if each $e \in E$ is covered at most once. The value of C is defined to be $|C|$ — the number of elements covered by C. Given a feasible multi-set $\mathcal{P}$, we denote by $f(\mathcal{P})$ the maximal value of a feasible partial cover $C \subseteq \mathcal{P} \times E$.

Lemma 1 *Given a feasible multi-set $\mathcal{P}$, a cover C of value $f(\mathcal{P})$ can be computed in polynomial time. In particular, we can establish whether $\mathcal{P}$ is a feasible solution to the set cover problem.*

Proof: For each $S \in \mathcal{S}$, let $m^{\mathcal{P}}(S)$ denote the number of copies of S that appear in $\mathcal{P}$. We build the following directed network. Let $G = (L, R, E')$ be the directed incidence graph of $\mathcal{P}$ and E, i.e., L contains a vertex for each copy of each set in $\mathcal{P}$: $L = \{v_i(S) |\ S \in \mathcal{S}, 1 \leq i \leq m^{\mathcal{P}}(S)\}$, $R = E$. For each $v_i(S) \in L$, $e \in R$, there is an edge $(v_i(S), e) \in E'$ of capacity 1 iff $e \in S$. Add a source vertex s and an edge $(s, v_i(S))$ of capacity $k(S)$ for each $S \in \mathcal{S}, 1 \leq i \leq m^{\mathcal{P}}(S)$. Add a sink vertex t an edge (e, t) of capacity 1 for each $e \in E$.

Consider the maximum flow in this network. The value of the flow is at least $f(\mathcal{P})$, since the optimal cover defines a feasible flow in the network. Also, the maximum flow in the network is integral, and thus it induces a feasible partial cover of the same value.

Clearly, $\mathcal{P}$ is a feasible solution to the set cover problem iff $f(\mathcal{P}) = |E|$. □

Since vertex cover with hard capacities is a special case of set cover with hard capacities, (where each vertex $v \in V$ can be viewed as a set whose elements are the edges adjacent to v), all above definitions as well as Lemma 1 can also be applied to the vertex cover problem.

3. Vertex Cover with Hard Capacities

In this section, we consider the unweighted capacitated vertex cover problem. We show a 3-approximation for the special case where for each $v \in V$, $m(v) = 1$. The algorithm can be extended to obtain a 3-approximation for the case where $m(v)$ is arbitrary. In section 4.1 we show that the weighted version is set-cover hard. The approximation algorithm for the capacitated set cover problem can be used for the weighted capacitated vertex cover problem.

We start with the following linear programming relaxation of the problem. For $v \in V$, let $x(v)$ be a variable indicating whether v belongs to the cover. For $e = (u, v) \in E$, let $y(e, v)$ be a variable indicating whether vertex v covers edge e. For each $v \in V$, $N(v)$ denotes the set of edges adjacent to v.

$$\min \quad \sum_{v \in V} x(v) \qquad \text{(UVC)}$$

s.t.

$$\begin{aligned} y(e,u) + y(e,v) &= 1 && \text{for all } e = (u,v) \in E \\ y(e,v) &\leq x(v) && \text{for all } e \in E, v \in e \\ \sum_{e \in N(v)} y(e,v) &\leq k(v) \cdot x(v) && \text{for all } v \in V \\ x(v), y(e,v) &\geq 0 && \text{for all } v \in V, e \in E \\ x(v) &\leq 1 && \text{for all } v \in V \end{aligned}$$

Lemma 2 *Let (x, y) be a feasible solution to (UVC), where x is integral. Then, there exists a feasible solution (x, z) to (UVC) where z is also integral and z can be computed in polynomial time (given x).*

Proof: Let $U = \{v \in V | x(v) = 1\}$. We use Lemma 1 to compute an (integral) cover z of the edges by vertices in U. Note that y induces a fractional flow of value $|E|$ in the network constructed in the proof of Lemma 1. Thus, $c(U) = |E|$, and therefore, in z, all the edges are covered. □

3.1. The Rounding Algorithm

Consider a fractional optimal solution (x, y) to (UVC). We show how to round this solution, obtaining a feasible solution (x', y') where x' is integral. By Lemma 2, x' induces a capacitated vertex cover. The rounding algorithm consists of three major steps.

Step 1: (Setting it up) We need the following definitions.

- For each $u \in U$, define $U = \{u | x(u) \geq \frac{1}{3}\}$ and $\overline{U} = V \setminus U$.
- Define E' to be the set of edges with one endpoint in U and the other endpoint in $\overline{U}$.
- For each $u \in V$, $N'(u) = E' \cap N(u)$.

- For each $u \in U$, define $\ell(u) = \sum_{e \in N'(u)} y(e,u)$, and $r(u) = \sum_{e=(u,v) \in N'(u)} y(e,v) = |N'(u)| - \ell(u)$. Note that the value of $\ell(u)$ denotes the total contribution of u to the coverage of edges in $N'(u)$, and $r(u)$ denotes the total contribution of vertices in $\overline{U}$ to the coverage of these edges.

- For each $u \in U$, define $\epsilon(u) = \frac{1}{x(u)} - 1$, and $h(u) = (1 - 2\epsilon(u))r(u)$. The meaning of these variables is explained below.

The constraints of (UVC) guarantee that each edge $e = (u,v) \in E$ has at least one endpoint in U: Since $y(e,u) + y(e,v) = 1$, $y(e,u) \leq x(u)$, and $y(e,v) \leq x(v)$, it follows that $x(u) \geq \frac{1}{3}$ or $x(v) \geq \frac{1}{3}$ must hold.

Consider a vertex $u \in U$. If edge $(u,v) \in E'$ exists ($v \in \overline{U}$), then since $y(e,v) \leq x(v) < \frac{1}{3}$, it follows that $x(u) \geq y(e,u) > \frac{2}{3}$. It also follows that for each $u \in U$, $\ell(u) \geq 2r(u)$, since $y(e,u) \geq 2y(e,v)$ for each $(u,v) \in E'$.

Our cover is going to consist of the vertices of U together with a subset $I \subseteq \overline{U}$, such that $U \cup I$ can fractionally cover all the edges. (By Lemma 2, $U \cup I$ is also an integral feasible vertex cover.)

First, we round up $x(u)$ to be equal to 1 for each vertex $u \in U$. As a result, u can increase its contribution to the coverage of the edges belonging to $N'(u)$ by a factor of $1/x(u)$, i.e., it can cover each edge $e = (u,v) \in N'(u)$ by the fraction $z(e,u) = \frac{y(e,u)}{x(u)}$. (Note that $z(e,u) \leq 1$, since $y(e,u) \leq x(u)$, and also $\sum_{e \in N(u)} z(e,u) \leq k(u)$ by the following constraint of (UVC): $\sum_{e \in N(u)} y(e,u) \leq k(u) \cdot x(u)$.) Since $x(u) \geq \frac{2}{3}$, $0 \leq \epsilon(u) \leq \frac{1}{2}$. Thus, the edges belonging to $N'(u)$ get a contribution from u of at least

$$\begin{aligned}\sum_{e \in N'(u)} \frac{y(e,u)}{x(u)} &= \sum_{e \in N'(u)} y(e,u)(1+\epsilon(u)) \\ &= \ell(u)(1+\epsilon(u)) \\ &\geq \ell(u) + 2r(u)\epsilon(u).\end{aligned}$$

To complete the fractional cover, we need an additional coverage of value $(1 - 2\epsilon(u))r(u) = h(u)$ from vertices belonging to $\overline{U}$, since $\ell(u) + r(u)$ suffices to cover $N'(u)$. Our goal in the next two steps is to find $I \subseteq \overline{U}$ such that for each $u \in U$, the vertices from I can contribute to $N'(u)$ at least $h(u)$.

Step 2: (Randomized rounding) Each vertex $v \in \overline{U}$ is independently chosen to be in I with probability equal to $3x(v)$. For each vertex $v \in I$, for each $e \in N'(v)$, define a new cover of edge e by vertex v: $z(e,v) = \frac{y(e,v)}{x(v)}$.

Step 3: (Altering the rounding) In this step we start with a feasible fractional solution (x', y') and iteratively alter it until x' becomes integral, while maintaining feasibility of (x', y'). We denote by P the vertices in U that are in "deficit", i.e.,

$$P = \left\{ u \in U | \sum_{e=(v,u) \in E', v \in I} z(e,v) < h(u) \right\}.$$

Our initial feasible solution (x', y') for (UVC) is defined as follows: If $v \in U \cup I$, then $x'(v) = 1$, otherwise $x'(v) = x(v)$. For $e = (u,v)$, $y'(e,v)$ and $y'(e,u)$ are defined as follows.

- If $u, v \in U$, then $y'(e,u) = y(e,u)$ and $y'(e,v) = y(e,v)$.

- If $u \in U \setminus P$: if $v \in I$, then $y'(e,v) = z(e,v)$, else $y'(e,v) = 0$. Set $y'(e,u) = 1 - y'(e,v)$. Note that since $u \notin P$, it has enough capacity to complete the cover of $N'(u)$.

- If $u \in P$: if $v \in I$, then $y'(e,v) = z(e,v)$ and $y'(e,u) = 1 - z(e,v)$. (Note that $y'(e,u) \leq y(e,u)$, since $z(e,v) \geq y(e,v)$). Else ($v \notin I$), set $y'(e,u) = y(e,u)$ and $y'(e,v) = y(e,v)$.

It is easy to see that (x', y') is a feasible solution for (UVC). We now show how to get rid of P by adding new vertices to I. We charge the cost of the new vertices added to I to the vertices of P.

PROCEDURE ELIMINATE.

While $P \neq \emptyset$:

1. Let $u \in P$, $e = (u,v) \in E'$, such that $v \in \overline{U} \setminus I$ (there must be at least one such v). Let $P' = \{w \in P \mid w \neq u,\ e' = (w,v) \in E'\}$.

2. Add v to I (set $x'(v) = 1$).

   Update the cover: For each $w \in P'$ where $e' = (v,w) \in E'$, set $y'(e',v) = z(e',v) = \frac{y(e',v)}{x(v)}$. Set $y'(e',w) = 1 - y'(e',v)$. Note that the value of $y'(e',w)$ can only decrease. Set $y'(e,v)$ to be the minimum between 1 and the remaining capacity of v (which must be at least $y(e,v)$). Set $y'(e,u) = 1 - y'(e,v)$.

   Update the set P: For each $w \in P$ for which $\sum_{e=(w,a):a \in I} y'(e,a) \geq h(w)$, remove w from P. Update the cover of $N'(w)$ as follows. For each $e = (b,w) \in E'$ such that $b \notin I$, set $y'(e,b) = 0$ and $y'(e,w) = 1$. Note that w has enough capacity to cover all such edges.

It is easy to see that feasibility is maintained after each iteration. The number of iterations of PROCEDURE ELIMINATE is bounded by $|\overline{U}|$, since $|I|$ is increased by one in each iteration. At the end, when P becomes empty, for each v with $x'(v) < 1$, we set $x'(v) = 0$. The final solution is $U \cup I$. The next theorem follows from the discussion.

Theorem 3 *The algorithm computes a feasible solution (x', y') to (UVC), where x' is integral.*

To obtain an integral capacitated vertex cover, we apply Lemma 2 to the solution (x', y').

3.2. Analysis

The analysis of the rounding is divided into two parts.

Charging scheme for Step (3) We show that we can charge the cost of adding vertices to I in PROCEDURE ELIMINATE to the vertices in P, such that each $u \in P$ pays at most $h(u) + 1$. Consider an iteration of PROCEDURE ELIMINATE. We charge the vertices of $P' \cup \{u\}$ for adding v to I. Each $w \in P'$, where $e' = (w, v) \in E'$, is going to pay $z(e', v)$ (which is exactly the contribution of v to the cover of e'). Vertex u is going to pay the remaining cost (if any remains), which is also at most the contribution of v to the cover of the edge (u, v). We now bound the total amount charged to $a \in P$. While a is still in P, in each iteration it pays exactly an amount equal to the coverage that edges in $N'(a)$ get from the newly added vertex v. Once the coverage of $N'(a)$ coming from vertices in I exceeds $h(a)$, a is removed from P. Therefore, in total a pays at most $h(a) + 1$.

Bounding the cost We now bound the total cost of the solution produced.

Claim 4 *Let $u \in U$ such that $r(u) \geq \frac{3}{4(1+\epsilon(u))^2}$. Then, the probability that $u \in P$ after Step (2) is at most $\frac{3}{4(1+\epsilon(u))^2 r(u)}$.*

Proof: Consider $e = (u, v) \in N'(u)$. We define the random variable $t_e = \begin{cases} z(e,v) & v \in I \\ 0 & \text{otherwise} \end{cases}$. Variables t_e are independent since there are no parallel edges in the graph. Note that:

- $u \in P$ iff $\displaystyle\sum_{e \in N'(u)} t_e < h(u)$.
- The expectation of $\sum_{e \in N'(u)} t(e)$ is:

$$\mu = \mathbf{Exp}\left[\sum_{e \in N'(u)} t_e\right] = \sum_{e=(v,u) \in N'(u)} 3z(e, v) \cdot x(v) = 3r(u)$$

- The variance of $\sum_{e \in N'(u)} t(e)$ is:

$$\sigma^2 = \mathbf{Var}\left[\sum_{e \in N'(u)} t_e\right] = \sum_{e=(u,v) \in N'(u)} z^2(e, v) \cdot 3x(v) \cdot (1 - 3x(v)) \leq \mu$$

It follows from Chebyshev's Inequality, when applied to the random variable $\sum_{e \in N'(u)} t_e$, that

$$\begin{aligned} &\mathbf{Prob}\left[\sum_{e \in N'(u)} t_e < h(u)\right] \\ &\leq \mathbf{Prob}\left[\left|\sum_{e \in N'(u)} t_e - \mu\right| \geq \mu - r(u)(1 - 2\epsilon(u))\right] \\ &= \mathbf{Prob}\left[\left|\sum_{e \in N'(u)} t_e - \mu\right| \geq 2r(u)(1 + \epsilon(u))\right] \\ &\leq \frac{\sigma^2}{4r^2(u)(1+\epsilon(u))^2} \\ &\leq \frac{3}{4r(u)(1+\epsilon(u))^2} \end{aligned}$$

□

We are now ready to compute the expected cost of the solution.

- For $v \in \overline{U}$, the expected cost we pay in Step (2) is $3x(v)$.
- For $u \in U$ where $N'(u) = \emptyset$, we pay at most $3x(u)$ in Step (1), and we do not pay in Step (3).
- For $u \in U$ where $N'(u) \neq \emptyset$, consider two cases:
  - If $r(u) \geq \frac{3}{4(1+\epsilon(u))^2}$, then in Step (1) we pay a unit for opening u. In Step (3), we pay at most

$h(u)+1$ with probability at most $\frac{3}{4r(u)(1+\epsilon(u))^2}$. Thus, the expected cost is bounded by

$$\begin{aligned}
&1+(h(u)+1)\cdot\frac{3}{4r(u)(1+\epsilon(u))^2}\\
&=1+\frac{3(1-2\epsilon(u))}{4(1+\epsilon(u))^2}+\frac{3}{4r(u)(1+\epsilon(u))^2}\\
&\leq 2+\frac{3(1-2\epsilon(u))}{4(1+\epsilon(u))^2}
\end{aligned}$$

– If $r(u) < \frac{3}{4(1+\epsilon(u))^2}$, then in Step (1) we pay a unit for opening u, and at most $h(u)+1$ in Step (3). In total we pay:

$$\begin{aligned}
2+h(u) &= 2+(1-2\epsilon(u))r(u)\\
&\leq 2+\frac{3(1-2\epsilon(u))}{4(1+\epsilon(u))^2}
\end{aligned}$$

In both cases it suffices to note that $2+\frac{3(1-2\epsilon(u))}{4(1+\epsilon(u))^2}\leq 3x(u)=\frac{3}{1+\epsilon(u)}$ holds for all $\epsilon(u)$, $0\leq\epsilon(u)\leq\frac{1}{2}$.

4. Hardness Results

4.1. Weighted Vertex Cover

We show that the capacitated vertex cover with arbitrary weights is at least as hard as the set cover problem. Given an instance of the set cover problem, let $G=(L,R,E')$ be its bipartite incidence graph, where $L=\mathcal{S}$, $R=E$, $(S,e)\in E'$ iff $e\in S$. For each vertex v in the graph, let $\delta(v)$ denote its degree. For each $v\in L$, define $w(v)$ to be the weight of the corresponding set, and $k(v)=\delta(v)$. For each $v\in R$, define $w(v)=0$, and $k(v)=\delta(v)-1$. For each vertex v in the graph, the multiplicity $m(v)=1$. Given a solution to the set cover instance, the solution to the capacitated vertex cover consists of all the vertices of R and the vertices from L corresponding to the sets in the set cover. The set vertices can cover all their adjacent edges. Since each element is covered in the set cover solution, for each $v\in R$, at least one of its adjacent edges is covered by a set vertex, so v has enough capacity to cover the remaining edges. The other direction is also true. Given a feasible solution to the vertex cover problem, we can find a feasible solution to the set cover problem of the same cost. The solution to the set cover problem consists of the sets corresponding to the vertices of L that participate in the solution of the vertex cover instance.

4.2. Vertex Cover with Unsplittable Demands

We assume that each edge e has a demand $d(e)$ that must be supplied by one of its endpoints. For each $v\in V$, the sum of the demands of the adjacent edges that v supplies must not exceed the capacity $k(v)$. It is impossible to approximate this problem, since, given a problem instance, it is NP-hard to answer the question whether V (the set of all the vertices in the problem instance) is a feasible vertex cover. The reduction is from partition. Suppose we have N numbers $n_1,\ldots,n_N$. We build two special vertices v and u with capacities $k(v)=k(u)=\frac{1}{2}\sum_{i=1}^{N}n_i$. Additionally, for each element n_i, we add a vertex i with capacity $k(i)=n_i$, and two edges: (v,i) and (u,i) with demands equal to n_i. If there is a feasible assignment of all the edges to the vertices, then the set of edges that u and v cover defines a partition, since for each i, either u covers the edge (u,i), or v covers the edge (v,i). The converse is also true: a partition trivially defines a feasible assignment of edges.

5. Set Cover with Hard Capacities

In this section we consider the set cover problem with hard capacities. For the sake of simplicity, we assume that for each set $S\in\mathcal{S}$, only one copy is available, i.e., $m(S)=1$. If this is not the case, we can view each available copy of each set as a distinct set. Consider the following greedy algorithm.

ALGORITHM GREEDY COVER:

Start with $\mathcal{P}=\emptyset$.

While $\mathcal{P}$ is not a feasible capacitated set cover:

> For each $S\in\mathcal{S}\setminus\mathcal{P}$, let $f(S)$ be the maximum possible increase in the number of elements covered by adding S to $\mathcal{P}$, i.e. $f(S)=f(\mathcal{P}\cup\{S\})-f(\mathcal{P})$. Among the sets in $\mathcal{S}\setminus\mathcal{P}$, let $S=\arg\min_{S:\ f(S)>0}\frac{w(S)}{f(S)}$. Add S to $\mathcal{P}$.

Note that the values $f(S)$ can be computed using Lemma 1. Wolsey [23] shows that ALGORITHM GREEDY COVER achieves a $\log(\max_S|S|)$-approximation, using the dual fitting technique. We show a simpler and a more intuitive charging scheme that proves the same result.

Let $\mathcal{T}\subseteq\mathcal{S}$ be a collection of sets, and let $C\subseteq\mathcal{T}\times E$ be a feasible partial cover. We will always assume that each element e is covered by at most one set $S\in\mathcal{T}$. For each $\mathcal{T}'\subseteq\mathcal{T}$, we denote by $f_C(\mathcal{T}')$ the number of elements covered by sets in $\mathcal{T}'$. We need the following result.

Lemma 5 *Let $\mathcal{T}$ be a feasible solution to the set cover problem and let $\mathcal{T}_1,\mathcal{T}_2$ be a partition of $\mathcal{T}$ into two disjoint subsets. Then there is a feasible cover $C\subseteq\mathcal{T}\times E$, such that all the elements are covered in C and $f_C(\mathcal{T}_1)=f(\mathcal{T}_1)$.*

Proof: Let $C\subseteq\mathcal{T}\times E$ be a feasible cover, where each element $e\in E$ is covered by some $S\in\mathcal{T}$, and assume that $f_C(\mathcal{T}_1)<f(\mathcal{T}_1)$. Let $C'\subseteq\mathcal{T}_1\times E$ be a feasible partial

cover, where $f_{C'}(\mathcal{T}_1) = f(\mathcal{T}_1)$. Note that the existence of cover C' follows from the definition of $f(\mathcal{T}_1)$. As C' is a partial cover, some elements may not be covered in C'. We gradually change the cover C, while maintaining its feasibility, until the condition of the lemma holds. Perform the following procedure:

While $f_C(\mathcal{T}_1) < f(\mathcal{T}_1)$:

> Let $S \in \mathcal{T}_1$ be a set, such that S covers less elements in C than it does in C'. There is at least one such set since $f_C(\mathcal{T}_1) < f(\mathcal{T}_1)$.
>
> Let $e \in E$ be some element covered by S in C', but covered by some $T \neq S$ in C. Note that it is possible that $T \in \mathcal{T}_1$, or $T \in \mathcal{T}_2$. Change C so that now e is covered by S, i.e., remove (T, e) and add (S, e) to C.

It is clear that we can perform the procedure and maintain a feasible cover C, while $f_C(\mathcal{T}_1) < f(\mathcal{T}_1)$. The moment a pair $(S, e) \in C'$ is added to C, it stays there till the end of the procedure. Thus, the number of iterations is bounded by $|C'|$ and is therefore finite. At the end of the procedure we have a cover that satisfies the conditions of the lemma. □

We now proceed with the analysis of ALGORITHM GREEDY COVER. Denote the solution computed by ALGORITHM GREEDY COVER by $\mathcal{P} = \{S_1, S_2, \ldots, S_k\}$, and assume that the sets are added to the solution by the algorithm in this order. For each i, $0 \leq i \leq k$, let $\mathcal{P}_i = \{S_1, S_2, \ldots, S_i\}$ be the solution at the end of iteration i. Let OPT be an optimal solution. We "replay" the algorithm, while charging the costs of the sets added to $\mathcal{P}$ by ALGORITHM GREEDY COVER to the sets in OPT.

Start with $\mathcal{P}_0 = \emptyset$. For each $S \in OPT$, let $a_0(S)$ be the number of elements covered by S in OPT (assuming every element is covered by exactly one set in OPT). For each iteration i of ALGORITHM GREEDY COVER, new values $a_i(S)$ of sets in $OPT \setminus \mathcal{P}_i$ are defined. The following invariant holds throughout the analysis: we can cover all the elements by the sets in $OPT \cup \mathcal{P}_i$, even if the capacities of sets $S \in OPT \setminus \mathcal{P}_i$ are restricted to be $a_i(S)$.

The invariant is clearly true for $\mathcal{P}_0$ and a_0. Consider iteration i of ALGORITHM GREEDY COVER. We add set S_i to the solution. Since the invariant holds for $\mathcal{P}_{i-1}$, a_{i-1}, the collection of sets $\mathcal{P}_i \cup OPT$ is a feasible cover, even if we restrict the capacities of sets $S \in OPT \setminus \mathcal{P}_i$ to be $a_{i-1}(S)$. By Lemma 5, there is a feasible cover $C \subseteq (\mathcal{P}_i \cup OPT) \times E$, where each set $S \in OPT \setminus \mathcal{P}_i$ covers at most $a_{i-1}(S)$ elements and the sets in $\mathcal{P}_i$ cover exactly $f(\mathcal{P}_i)$ elements. For each $S \in OPT \setminus \mathcal{P}_i$, define $a_i(S)$ to be the number of elements covered by S in C. Note that $a_i(S) \leq a_{i-1}(S)$.

If $S_i \in OPT$, we do not charge any sets for its cost, since OPT also pays for it. Otherwise, suppose $f(S_i) = n_i$. The number of elements covered by sets in $OPT \setminus \mathcal{P}_i$ in C is $\sum_{S \in OPT \setminus \mathcal{P}_i} a_{i-1}(S) - n_i$. Therefore, $\sum_{S \in OPT \setminus \mathcal{P}_i} (a_{i-1}(S) - a_i(S)) = n_i$. We charge each $S \in OPT \setminus \mathcal{P}_i$ with $\frac{w(S_i)}{n_i} \cdot (a_{i-1}(S) - a_i(S))$. Note that the total cost charged to the sets in OPT in this iteration is exactly $w(S_i)$.

We now bound the cost charged to each $S \in OPT$. If $S \in \mathcal{P}$, let j denote the iteration when S is added to $\mathcal{P}$. Otherwise, let $j = k + 1$, where k is the total number of iterations. For each $i < j$, at the beginning of iteration i, $f(S) \geq a_{i-1}(S)$. This follows from the way the value of $a_{i-1}(S)$ is determined. Since ALGORITHM GREEDY COVER chooses a set other than S in this iteration, $\frac{w(S_i)}{n_i} \leq \frac{w(S)}{a_{i-1}(S)}$. The total value charged to S is:

$$
\begin{aligned}
&\sum_{i=1}^{j-1} (a_{i-1}(S) - a_i(S)) \frac{w(S_i)}{n_i} \\
&\leq w(S) \sum_{i=1}^{j-1} \frac{(a_{i-1}(S) - a_i(S))}{a_{i-1}(S)} \\
&\leq w(S) H(a_0(S)) \\
&\leq w(S) H(|S|)
\end{aligned}
$$

The total cost of the solution is bounded by $OPT(1 + \ln(\max_S |S|))$.

6. Extensions

It is not hard to show that the natural ilinear programs for the set cover problem with hard capacities, as well as the more general submodular set cover problem, have an unbounded integrality gap. Wolsey [23] shows using the dual fitting technique that ALGORITHM GREEDY COVER achieves an $O(\log(f_{\max}))$-approximation for the general submodular set cover problem, where $f_{\max} = \max_{S \in \mathcal{S}} f(\{S\})$. Wolsey uses the following linear programming formulation:

$$
\min \quad \sum_{S \in \mathcal{S}} w(S) x(S) \qquad \text{(SSC)}
$$

s.t.

$$
\begin{aligned}
\sum_{S \in \mathcal{T}} f_{\mathcal{T}}(S) x(S) \ &\geq\ f(\mathcal{S} \setminus \mathcal{T}) \qquad \forall \mathcal{T} \subseteq \mathcal{S} \\
x(S) \ &\geq\ 0 \qquad \forall S \in \mathcal{S}
\end{aligned}
$$

Here, $f_{\mathcal{T}}(S) = f(\mathcal{T} \cup \{S\}) - f(\mathcal{T})$.

We remark that our analysis of Algorithm Greedy Cover can be extended to prove a similar approximation guarantee for the submodular set cover problem.

An interesting special case of the submodular set cover problem is the multi-set multi-cover problem. In this problem, the input sets are actually multi-sets, i.e. an element $e \in E$ can appear in $S_j \in \mathcal{S}$ more than once, and the elements have splittable demands. The multi-set multi-cover problem with unbounded set capacities can be defined as an IP: $\min\{w^T x \mid Ax \geq d, 0 \leq x \leq b, x \in Z\}$. The constraints $x \leq b$ are called multiplicity constraints, and they generally make covering problems much harder, as the natural linear programming relaxation has an unbounded integrality gap. Dobson [8] gives a combinatorial greedy $H(\max_{1 \leq j \leq m} \sum_{1 \leq i \leq n} A_{ij})$-approximation algorithm, where $H(t)$ is the tth harmonic number. This is a logarithmic approximation factor for the case where A is a $\{0, 1\}$ matrix (set multi-cover), but can be as bad as a polynomial approximation bound in the general case (multi-set multi-cover). Recently, Carr, Fleischer, Leung and Phillips [4] gave a p-approximation algorithm, where p denotes the maximum number of variables in any constraint. Their algorithm is based on a linear relaxation in the spirit similar to that of (SSC). Using similar ideas for strengthening the linear program, Kolliopoulos and Young [16] obtained an $O(\log n)$-approximation.

For the multi-set multi-cover problem with hard capacities, the function $f(\mathcal{T})$ can be computed in polynomial time. Thus, ALGORITHM GREEDY COVER can be implemented to run in polynomial time, achieving an approximation ratio of $O(\log(\max_{S \in \mathcal{S}} |S|))$.

References

[1] E. Balas and M Padberg. Set partitioning: a survey. *SIAM Review*, 18:710–760, 1976.

[2] J. Bar-Ilan, G. Kortsarz, D, Peleg. Generalized submodular cover problems and applications. In *Proceedings of the 4th Israel Symposium on Computing and Systems* 1996, pp. 110-118.

[3] 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.

[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 *Proceedings of the 11th ACM-SIAM Symposium on Discrete Algorithms*, pp. 106–115, 2000.

[5] M. Charikar, S. Guha, E. Tardos and D. Shmoys. A constant-factor approximation algorithm for the k-median problem. In *Proceedings of the 31st Annual ACM Symposium on the Theory of Computing*, pp. 1-10, 1999.

[6] N. Christofides and S. Korman. A computational survey of methods for the set covering problem. *Management Science* 21:591–599, 1975.

[7] V. Chvátal. A greedy heuristic for the set-covering problem. *Mathematics of Operations Research*, 4(3):233–235, 1979.

[8] G. Dobson. Worst-case analysis of greedy heuristics for integer programming with non-negative data. *Mathematics of Operations Research*, 7(4): 515-531, 1972.

[9] U. Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45(4):634–652, July 1998.

[10] R. Gandhi, S. Khuller, S. Parthasarathy, A. Srinivasan. Dependent rounding in bipartite graphs. In *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science*, 2002.

[11] R.S. Garfinkel and G.L. Nemhauser. Optimal set covering: a survey. In *Perspectives on optimization: a collection of expository articles*, A.M. Geofrion, ed., 164–183, 1972.

[12] S. Guha, R. Hassin, S. Khuller, and E. Or. Capacitated vertex covering with applications. In *Proceedings of the 13th Symposium on Discrete Algorithms*, pp. 858–865, 2002.

[13] D.S. Hochbaum. Approximation algorithms for the set covering and vertex cover problems. *SIAM Journal on computing*, 11:555-556, 1982.

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

[15] D. S. Johnson. Approximation algorithms for combinatorial problems. *J. Comput. System Sci.*, 9:256-278, 1974.

[16] S.G. Kolliopoulos and N.E. Young. Tight approximation results for general covering integer programs. In *Proceedings of the 42nd Annual Symposium on Foundations of Computer Science*, pp. 522–528, 2001.

[17] L. Lovász. On the ratio of optimal and fractional covers. *Discrete Mathematics*, 13:383-390, 1975.

[18] P. B. Mirchandani and R. L. Francis (editors). Discrete location theory. Wiley Interscience, 1990.

[19] M.W. Padberg. Covering, packing and knapsack problems. *Annals of Discrete Mathematics* 4:265–287, 1979.

[20] D. B. Shmoys, É. Tardos, and K. Aardal. Approximation algorithms for the facility location problem. In *Proceedings of the 29th Annual ACM Symposium on the Theory of Computing*, pp. 265-274, 1997.

[21] M. Pál, É. Tardos, T. Wexler. Facility location with nonuniform hard capacities. In *Proc. 42nd Annual Symposium on Foundations of Computer Science*, pp. 329–338, 2001.

[22] V. V. Vazirani. Approximation algorithms. Springer-Verlag, 2001.

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