Algorithms for Hub Label Optimization

Maxim Babenko[1], Andrew V. Goldberg[2],
Anupam Gupta[3], and Viswanath Nagarajan[4]

[1] Department of Mechanics and Mathematics, Moscow State Univeriity; Yandex.
[2] Microsoft Research Silicon Valley
[3] Carnegie Mellon University and Microsoft Research SVC.
[4] IBM T.J. Watson Research Center.

Abstract. Cohen et al. developed an $O(\log n)$-approximation algorithm for minimizing the total hub label size (ℓ_1 norm). We give $O(\log n)$-approximation algorithms for the problems of minimizing the maximum label (ℓ_∞ norm) and minimizing ℓ_p and ℓ_q norms simultaneously.

1 Introduction

Modern applications, such as computing driving directions and other location-based services, require very fast point-to-point shortest path algorithms. Although Dijkstra's algorithmsolves this problem in near-linear time [14] on directed and in linear time on undirected graphs [16], some applications require sublinear distance queries. This motivates preprocessing-based algorithms, which yield sublinear queries on some graph classes (e.g., [10, 13]). In particular, Gavoille et al. [13] introduced *distance labeling* algorithms. These algorithms precompute *labels* for each vertex such that the distance between any two vertices s and t can be computed using only their labels.

A prominent case of this paradigm is *hub labeling (HL)*: the label of v consists of a collection of vertices (the *hubs* of v) with their distances from v. Hub labels satisfy the *cover property*: for any two vertices s and t, there exists a vertex w on the shortest s–t path that belongs to both the label of s and the label of t. Given this information, distance queries are easy to implement: for two vertices v and w, we compute the sums of the v-u and u-w distances over vertices u in the intersection of the labels of v and w, and return the minimum value found.

Cohen at al. [9] gave an $O(\log n)$-approximation algorithm for the smallest-size labeling, where n denotes the number of vertices and the size of the labeling is the sum of the number of hubs in the vertex labels. (This also minimizes the average label size.) The algorithm uses an elegant reduction to the set-cover problem [8]. At each step, the algorithm solves a maximum density subgraph problem, which can be done exactly using parametric flows [12] or by a faster approximation algorithm [15]. HL leads to the fastest implementation of the point-to-point shortest path queries in road networks [1], and works well on some other network types [2]. This motivates further theoretical study of HL.

In this paper we consider approximation algorithms for the optimization problem of producing small labels. Since minimizing the average label size may potentially lead to imbalanced solutions where the label of some vertices are relatively large, a natural objective is to minimize the maximum HL size, which determines the worst-case query time. We give a polynomial time algorithm that finds an $O(\log n)$ approximation of the maximum label size, where n is the size of the graph. Our algorithm is based on reducing the HL problem to a non-standard set covering problem, where the objective is to cover all the elements using sets in such a way that no element is covered too often. Our algorithm is combinatorial, and is based on exponential cost functions, as in [4] and many other contexts.

If we consider a vector whose components correspond to the number of vertex hubs, then the total label size is the $\ell_1$norm of this vector and the maximum label size is the ℓ_∞ norm. This brings a natural generalization of the above problems, that of optimizing the ℓ_p norm of this vector. Our second result is an $O(\log n)$-approximation algorithm for this more general problem. This is also a combinatorial algorithm, where we naturally use degree-p polynomials instead of exponential cost functions, as in [5].

In applications, there are multiple criteria that one would like to be good for—one wants to simultaneously minimize the total label size (i.e., the space needed to store the labels) and the maximum label size (i.e., the worst-case query time). This is a bi-criteria optimization problem, with the optimal solutions on the Pareto-optimal curve. E.g., suppose that there is a labeling with total label size T_1 and maximum label size T_∞ (e.g., a labeling on the Pareto curve). Our third result is a polynomial-time algorithm to find a labeling with the total label size $O(T_1 \log n)$ and the maximum label size $O(T_\infty \log n)$. In fact, our techniques easily extend to a more general result: a logarithmic approximation to the problem of maintaining k moments of the label sizes. Specifically, given any set $P = \{p_1, p_2, \ldots, p_k\}$, and values T_i such that there exists a labeling whose ℓ_{p_i} norm is at most T_i for each $p_i \in P$, we can find a labeling whose ℓ_{p_i} norm is at most $O(k \log n) \cdot T_i$ for all $p_i \in P$.

1.1 Related Work

There is much work on minimizing multiple norms of a vector. For some problems, one can find a single solution that is simultaneously good against the best solution for each ℓ_p norm individually. E.g., Azar et al. [7] considered the restricted-assignment machine scheduling problem, and gave a solution which 2-approximates the best solution for each ℓ_p norm; this was extended and improved by [6, 3]. In some settings, it is possible to get better bounds for ℓ_p norm minimization for different values of p: e.g., [5] show p-competitive online algorithms for minimizing the ℓ_p norms of machine loads. However, the low-load set covering problem has a hardness of $\Omega(\log n)$ for all p, which means new techniques would be needed to get better approximations for HL.

2 Definitions and Notation

In the HL problem, we are given a graph $G = (V, E)$ with a distinguished shortest path P_{ij} between each pair of vertices $i, j \in \binom{V}{2}$. A *hub labeling* (HL) is an assignment of labels $L_i \subseteq V$ for each vertex $i \in V$ such that for any $i, j \in V$, we have some vertex $u \in L_i \cap L_j$ that lies on the path P_{ij}. For the purposes of this paper, the graph can be directed or undirected, and the path P_{ij} can be an arbitrary path—we will not use the fact that it is a shortest i-j path. If we consider the vector $\mathbf{L} = (L_1, L_2, \ldots, L_n)$, then we are interested in finding labelings with small ℓ_p norms: $\|\mathbf{L}\|_p := (\sum_{i=1}^{n} L_i^p)^{1/p}$ and $\|\mathbf{L}\|_\infty := \max_{i \in V} L_i$. We assume $p \in [1, \log n]$, since $\ell_{\log n}$ approximates all higher ℓ_p norms to within constant factors.

We will reduce HL to a low-load set-covering problem (LSC), which is defined as follows. As in the usual set cover problem, we are given a set system $(U, \mathcal{F})$, where $\mathcal{F} = \{S_1, S_2, \ldots\}$ is a collection of subsets of the universe U with N elements. A sub-collection $\mathcal{C} \subseteq \mathcal{F}$ is a *set cover* if $\cup_{S \in \mathcal{C}} S = U$, every element of U is contained in some set in $\mathcal{C}$. The elements in U are either *relevant* (denoted by $R \subseteq U$) or *irrelevant* (those in $U \setminus R$): for any relevant element $e \in R$, let $A_{\mathcal{C}}(e) = \#\{S \in \mathcal{C} \mid e \in S\}$ be the *load* of element e under this set cover $\mathcal{C}$, the number of sets in $\mathcal{C}$ that contain e. (Imagine the irrelevant elements to always have load 0.) For any $p \in [1, \infty)$, the ℓ_p norm of the loads is $\|A_{\mathcal{C}}\|_p = (\sum_{e \in R} A_{\mathcal{C}}(e)^p)^{1/p}$; the ℓ_∞ norm is $\|A_{\mathcal{C}}\|_\infty = \max_{e \in R} A_{\mathcal{C}}(e)$. To reiterate: we have to cover all the elements, relevant or otherwise, but we count the load only for the relevant elements.

Our algorithms use approximate max-density oracles. A *max-density oracle* takes costs c_e for relevant elements $e \in R$ and a set $X \subseteq U$ of elements already covered, and outputs a set $S \in \mathcal{F}$ that minimizes $\frac{\sum_{e \in S \cap R} c_e}{|S \setminus X|}$. In Section 3.2 we show how to implement the oracle for LSC instances arising from HL.

3 Application to Shortest Path Labels

Our motivating application is HL ℓ_p norm optimization. We show a reduction from the label optimization problem to LSC and an implementation of an approximate max-density oracle for the corresponding LSC problem.

3.1 From Labels to Set Covers

We model an instance $\mathcal{I}$ of HL as the following instance $\mathcal{I}'$ of LSC.

- The elements are all $\{i, j\}$ pairs, and all vertices — i.e., $U := \binom{V}{2} \cup V$. The elements of V are relevant and the elements of $\binom{V}{2}$ are irrelevant.
- For each vertex $x \in V$, let Q_x be the set of pairs $\{i, j\} \in \binom{V}{2}$ such that $x \in P_{ij}$. For any set of pairs $Q \subseteq \binom{V}{2}$, let $V(Q)$ denote the vertices that lie in at least one of these pairs—i.e., $V(Q) = \cup_{\{i,j\} \in Q} \{i, j\}$. Now for each $x \in V$, for each $Q \subseteq Q_x$, add the set $Q \cup V(Q)$ to the collection $\mathcal{F}$. Note that this may give us exponentially many sets.
- The problem is to compute a set cover $\mathcal{C} \subseteq \mathcal{F}$ minimizing the ℓ_p norm $\|A_{\mathcal{C}}\|_p$ (which only involves the load on relevant elements, i.e. vertices).

Lemma 1. *Minimizing the ℓ_p norm of an instance $\mathcal{I}$ of HL is equivalent to minimizing the ℓ_p norm of the corresponding instance $\mathcal{I}'$ of LSC.*

Proof. Given a solution for $\mathcal{I}$, construct a solution for $\mathcal{I}'$ as follows. For each $x \in V$, take $S_x \subseteq \binom{V}{2}$ to be the pairs that x "covers" in this solution—i.e., pairs $\{i, j\}$ such that x lies in $L_i \cap L_j$ and also on the path P_{ij}. Picking the sets $S_x \cup V(S_x)$, one for each x, we get a cover $\mathcal{C}$ for $\mathcal{I}'$. If the label of i does not contain x, then $i \notin V(S_x)$. Thus only the vertices x in the label of i can contribute to $A_{\mathcal{C}}(i)$, and therefore $A_{\mathcal{C}}(i) \leq |L_i|$.

Given a solution $\mathcal{C}$ to $\mathcal{I}'$, we construct a label L for $\mathcal{I}$ as follows. If $\mathcal{I}'$ contains a set $(Q \cup V(Q))$ for some $Q \subseteq Q_x$, then add x to L_i for all $i \in V(Q)$. Since all pairs in U are covered by $\mathcal{C}$, for each pair $\{i, j\}$, the labels L_i and L_j intersect at some vertex of P_{ij}. For every vertex i, we add x to L_i only if there is a set $(Q \cup V(Q)) \in \mathcal{C}$ with $Q \subseteq Q_x$ that covers i (i.e. $i \in V(Q)$). Thus $|L_i| \leq A_{\mathcal{C}}(i)$. □

3.2 Max-Density Oracle

In this section we show how to construct approximate max-density oracles for the LSC instances $\mathcal{I}'$ obtained via a reduction from HL. Recall that there are costs $c_i \geq 0$ for all $i \in V$ (relevant elements). Let us divide U into U_P (the pairs in V), and U_V (the vertices in V); recall that U_V are the relevant elements. For any set $S \subseteq U$, we now use $S_P := S \cap U_P$ and $S_V := S \cap U_V$.

Lemma 2. *Solving* $\min_{S \in \mathcal{F}} \frac{\sum_{v \in S_V} c_v}{|S_P \setminus X_P|}$ *gives a 3-approximate max-density oracle.*

Proof. For any $T \subseteq U$, we have $T = T_P \cup T_V$. By the structure of our sets, if some pair $\{i, j\} \in T_P$ then it must be the case that $\{i, j\} \subseteq T_V$. Hence, $V(T_P) \subseteq T_V$. Moreover, $|T_P \cup V(T_P)| \leq 3|T_P|$, since each pair in T_P can contribute at most both its endpoints to $V(T_P)$. Combining these facts, we get $|T_P| \leq |T| \leq 3|T_P|$. This implies the lemma. □

Recall that the Cohen et al. [9] algorithm approximates the ℓ_1 norm of the labels. Their algorithm uses a subroutine for the *maximum density subgraph* problem. We use a subroutine for the *weighted* variant of the problem: given a non-negative cost function $c : V \to \mathcal{R}_+$, the *density* of a graph H is $\mu(H) = \frac{|E(H)|}{c(V(H))}$. Note that $\mu(H)$ is undefined if $c(V(H)) = 0$. In this case if $|E(H)| = 0$ we define $\mu(H) = 0$, and otherwise $\mu(H) = \infty$. The *maximum density subgraph* problem is to find a vertex-induced subgraph of maximum density. This can be solved exactly using network flow [12]. In the full version, we describe a generalization of the faster 2-approximation algorithm [15] for the unit-weight maximum density subgraph to the weighted case.

Fix $v \in V$ and a set $X \subseteq U$ of covered elements. We define the v-center graph G_v as follows. The vertex set of G_v is V. Two vertices i, j are connected in G_v iff the pair $\{i, j\} \notin X$ and $P_{i,j} \ni v$, i.e., if v covers the pair.

Lemma 3. *We can reduce* $\min_{S \in \mathcal{F}} \frac{\sum_{v \in S_V} c_v}{|S_P \setminus X_P|}$ *to n weighted maximum density subgraph problems.*

Proof. Let S' be some set of vertices. Suppose that we add a vertex v to the labels of all vertices in S'. Then the edges in the subgraph of G_v induced by S' correspond to previously uncovered $\{i, j\}$ pairs that become covered. So the maximum density subgraph of G_v yields the set that minimizes the desired expression over the sets that correspond to v. The result follows by minimizing over all v (i.e., solving n weighted maximum density subgraph problems). □

A ρ-approximate solution to the weighted maximum subgraph problem gives a ρ-approximate minimum. Note that the results of this section extend to directed graphs. In this case the center graphs will become bipartite, as in [9].

4 The ℓ_∞ case: Minimizing the Maximum Load

In this section we investigate the problem of finding LSC $\mathcal{C}$ that approximately minimizes the maximum load of any relevant element in U. Recall that we have to cover the irrelevant elements (those in $U \setminus R$), even though we do not care about the load on them. Suppose that we know the optimal load $u = \|A_{\mathcal{C}^*}\|_\infty$; we can enumerate over all the possible values of u. We show a combinatorial greedy-like algorithm that achieves $\rho = O(\log N)$, where $N = |U|$.

Since the family $\mathcal{F}$ may consist of an exponential number of sets, and it may not be possible to look over all sets to find the best one, we assume that we have an α-approximate *max-density oracle* through which we access the set system.

Theorem 1. *There exists a combinatorial algorithm that makes at most n calls to an α-approximate max-density oracle, and finds a set cover $\mathcal{C}$ with element loads $A_\mathcal{C}(e) \leq O(\alpha \log N) \cdot \|A_{\mathcal{C}^*}\|_\infty$.*

The algorithm is a multiplicative-weights algorithm. It proceeds in *rounds*, where one set is added in each round. Let $\varepsilon = 1/(8\alpha)$. Let $A_t(e)$ be the number of times a relevant element $e \in R$ has been covered at the beginning of round t; at the very beginning of the process we have $A_1(e) = 0$ for all $e \in R$. Define the round-t cost of elements $e \in R$ as $c_t(e) := (1+\varepsilon)^{A_t(e)/u} \cdot \big((1+\varepsilon)^{1/u} - 1\big)$; so the round-$t$ cost of set S is $c_t(S) := \sum_{e \in S \cap R} (1+\varepsilon)^{A_t(e)/u} \cdot \big((1+\varepsilon)^{1/u} - 1\big)$.

Note the sum is only over the relevant elements in S. The algorithm is simple: consider the beginning of round t, when $t-1$ sets have already been picked, and let X_t be the elements already covered at this time. (Hence $X_1 = \emptyset$.) If all elements have not yet been covered (i.e., if $X_t \neq U$), use the α-approximate max-density oracle with costs $c_t(S)$ and the set X_t to obtain the next set.

For the analysis, define the potential at the beginning of round t to be $\Phi(t) := \sum_{e \in R} (1+\varepsilon)^{A_t(e)/u}$. The potential at the beginning of round 1 is $\Phi(1) = N$.

Lemma 4. *If we pick set S in round t, then $\Phi(t+1) - \Phi(t) = c_t(S)$.*

Proof. By the definition of the potential,

$$\Phi(t+1) - \Phi(t) = \sum_{e \in R} (1+\varepsilon)^{A_{t+1}(e)/u} - \sum_{e \in R} (1+\varepsilon)^{A_t(e)/u}$$

$$= \sum_{e \in S \cap R} (1+\varepsilon)^{(A_t(e)+1)/u} - (1+\varepsilon)^{A_t(e)/u} = \sum_{e \in S \cap R} (1+\varepsilon)^{A_t(e)/u} \Big((1+\varepsilon)^{1/u} - 1 \Big)$$

which is $c_t(S)$ by the definition of the round-t cost. □

Lemma 5. *At any round t, there exists a set S with cost-to-coverage ratio* $\frac{c_t(S)}{|S \setminus X_t|} \leq \frac{\Phi(t)}{|U \setminus X_t|} \cdot 2\varepsilon$.

Proof. The round-t cost of all the sets in the set cover $\mathcal{C}^*$ is

$$\begin{aligned} \sum_{S \in \mathcal{C}^*} c_t(S) &= \sum_{S \in \mathcal{C}^*} \sum_{e \in S \cap R} (1+\varepsilon)^{A_t(e)/u} \cdot \Big((1+\varepsilon)^{1/u} - 1 \Big) & (1) \\ &= \sum_{e \in R} \left[(1+\varepsilon)^{A_t(e)/u} \cdot \Big((1+\varepsilon)^{1/u} - 1 \Big) \cdot \sum_{S \in \mathcal{C}^* : e \in S} 1 \right] \\ &\leq \sum_{e \in R} (1+\varepsilon)^{A_t(e)/u} \cdot 2\varepsilon/u \cdot u \quad \leq \quad \Phi(t) \cdot 2\varepsilon. & (2) \end{aligned}$$

(The equality in (1) uses the definition of $c_t(S)$, and (2) used $u \geq 1$ and $\varepsilon \leq 1/4$ and hence $(1+\varepsilon)^{1/u} \leq 1 + (\varepsilon/u)(1 + \varepsilon + \varepsilon^2 + \cdots) \leq 1 + 2\varepsilon/u$.) Since $|U \setminus X_t|$ universe elements are not yet covered, and we could have chosen all the sets in $\mathcal{C}^*$ to cover these remaining elements at cost $\sum_{S \in \mathcal{C}^*} c_t(S)$, there exists some set whose cost-to-coverage-ratio is at most $\Phi(t) \cdot 2\varepsilon/(|U \setminus X_t|)$. □

Let us partition the rounds into phases: phase i begins in round t if the number of uncovered elements is at most $N/2^i$ for the first time at the beginning of round t. Hence phase 0 begins with round 1 (when the number of uncovered elements is $N/2^0 = N$) and ends at the point we have covered half the elements, etc. Note that some phases contain no rounds at all.

Lemma 6. *If rounds a and b are the first and last rounds of some phase i, then* $\Phi(b+1) \leq 2 \cdot \Phi(a)$.

Proof. Consider the beginning of some round $t \in \{a, a+1, \ldots, b\}$ in phase i. By Lemma 5 there exists a set with cost-to-coverage-ratio is at most $\Phi(t) \cdot 2\varepsilon/(|U \setminus X_t|)$. Moreover, the α-approximate max-density oracle finds a set whose cost-to-coverage ratio at most α times as much; i.e., at most $2\alpha\varepsilon \frac{\Phi(t)}{|U \setminus X_t|} \leq \frac{1}{4} \frac{\Phi(t)}{|U \setminus X_t|}$, using the definition of $\varepsilon = 1/(8\alpha)$.

Since the potential $\Phi(t)$ is non-decreasing, and $|U \setminus X_t| \geq N/2^{i+1}$ in this phase, this last expression is at most $\frac{1}{4} \frac{\Phi(b+1)}{N/2^{i+1}}$. Moreover, we cover at most $N/2^i$ elements in this phase, so the total cost incurred is at most $\frac{1}{2}\Phi(b+1)$. By Lemma 4, the total cost incurred in the phase equals the change in potential, so $\Phi(b+1) - \Phi(a) \leq (1/2) \cdot \Phi(b+1)$. This proves the lemma. □

Proof of Theorem 1: We claim the potential at the end of the algorithm is at most N^2. Indeed, using Lemma 6, the potential at most doubles in each phase, whereas the number of yet-covered elements at least halves. Hence, at the end of phase $\log_2 N$, there are strictly less than $N/2^{\log_2 N} = 1$ elements (i.e., zero elements) remaining; the potential is at most $2^{\log_2 N} \cdot \Phi(1) = N^2$.

Suppose the last round in which we pick a set is $f-1$. Then for the final potential to be at most N^2, it must be the case that for each $e \in U$, $(1+\varepsilon)^{A_f(e)/u} \leq N^2$. This means that $A_f(e) \leq u \log_{1+\varepsilon}(N^2) = O(\log N)u/\varepsilon = O(\alpha \log N)u$. □

5 Simultaneous ℓ_1 and ℓ_∞ Norm Approximation

We can extend the results of Section 4 to find a set cover that simultaneously has small maximum load and small average load. In this case, suppose we are given non-negative values T and u such that there exists a set cover $\mathcal{C}^*$ with $\sum_{e \in R} A_{\mathcal{C}^*}(e) \leq T$, and also $A_{\mathcal{C}^*}(e) \leq u$ for all relevant elements $e \in R$. We want to find a cover $\mathcal{C}$ such that $\|A_{\mathcal{C}}\|_1 \leq T \cdot O(\alpha \log N)$, and $\|A_{\mathcal{C}}\|_\infty \leq u \cdot O(\alpha \log N)$.

For the algorithm, we use definitions of $c_t(e)$, $c_t(S)$, $\Phi(t)$, etc., from Section 4, but redefine ε to be $\frac{1}{24\alpha}$. We define the d-cost as $d_e = 1$ for $e \in R$; so $d(S) := |S \cap R|$ for any set S. The algorithm changes as follows: in round t, we now use the α-approximate density oracle to pick a set S (approximately) minimizing the "combined" ratio

$$\frac{c_t(S) + \varepsilon(\Phi(t)/T) \cdot d(S)}{|S \setminus X_t|}. \tag{3}$$

For the analysis, consider the beginning of round t, and call a set S *t-light* if $\frac{d(S)}{|S \setminus X_t|} \leq \frac{2T}{|U \setminus X_t|}$, and *$t$-heavy* otherwise. Let $\mathcal{C}_h^*$ denote the t-heavy sets in $\mathcal{C}^*$, and $\mathcal{C}_l^* := \mathcal{C}^* \setminus \mathcal{C}_h^*$ the light sets.

Lemma 7. *The number of elements from $U \setminus X_t$ covered by t-heavy sets in $\mathcal{C}^*$ is at most $\frac{1}{2}|U \setminus X_t|$.*

Proof. The fraction of the remaining elements that any t-heavy set S covers is $\frac{|S \setminus X_t|}{|U \setminus X_t|} \leq \frac{d(S)}{2T}$. Hence, the total fraction of remaining elements that t-heavy sets in $\mathcal{C}^*$ cover is $\sum_{S \in \mathcal{C}_h^*} \frac{|S \setminus X_t|}{|U \setminus X_t|} \leq \sum_{S \in \mathcal{C}_h^*} \frac{d(S)}{2T} \leq 1/2$. The last inequality is because $\sum_{S \in \mathcal{C}_h^*} d(S) \leq \sum_{S \in \mathcal{C}^*} |S \cap R| \leq T$. □

We can now modify Lemma 5 to say the following:

Lemma 8. *At any round t, there exists a t-light set S with cost-to-coverage ratio $\frac{c_t(S)}{|S \setminus X_t|} \leq \frac{\Phi(t)}{|U \setminus X_t|} \cdot 4\varepsilon$.*

Proof. Let $z := |U \setminus X_t|$ denote the number of universe elements not yet covered. Choosing all t-light sets in $\mathcal{C}^*$ at cost $\sum_{S \in \mathcal{C}_l^*} c_t(S)$, we would cover at least $z/2$ elements (by Lemma 7), hence there exists some set whose cost-to-coverage-ratio is at most $\frac{2}{z} \cdot \sum_{S \in \mathcal{C}_l^*} c_t(S) \leq \frac{2}{z} \cdot \sum_{S \in \mathcal{C}^*} c_t(S) \leq \frac{2}{z} \cdot \Phi(t) \cdot 2\varepsilon$. using the calculations as in Lemma 5. □

By Lemma 8, we infer there exists a set S whose combined cost-per-coverage is

$$\frac{c_t(S) + \varepsilon(\Phi(t)/T) \cdot d(S)}{|S \setminus X_t|} \le \frac{\Phi(t) \cdot 4\varepsilon + \varepsilon(\Phi(t)/T) \cdot 2T}{|U \setminus X_t|} = \frac{\Phi(t) \cdot 6\varepsilon}{|U \setminus X_t|}.$$

Our α-approximate density oracle finds a set S_t with combined-cost-per-coverage at most $\alpha \cdot 6\varepsilon \cdot \Phi(t)/|U \setminus X_t| = \frac{1}{4}\Phi(t)/|U \setminus X_t|$. Since the combined cost is a sum of non-negative quantities, we get

$$\frac{c_t(S_t)}{|S_t \setminus X_t|} \le \frac{1}{4}\frac{\Phi(t)}{|U \setminus X_t|} \qquad \text{and} \qquad \frac{d(S_t)}{|S_t \setminus X_t|} \le \frac{1}{4\varepsilon}\frac{T}{|U \setminus X_t|}. \tag{4}$$

The bound on $c_t(S_t)/|S_t \setminus X_t|$ can be used in the same fashion as in Section 4 to show that the potential function at most doubles during a phase, and there are at most $\log_2 N$ phases, so the maximum load is at most $O(\alpha \log N) \cdot u$. Moreover, the d-cost incurred in any phase i is at most $(N/2^i) \cdot \frac{1}{4\varepsilon} \cdot \frac{T}{N/2^{i+1}} = \frac{1}{2\varepsilon} \cdot T$. Summing over all $\log_2 N$ phases, and using $\varepsilon = 1/(24\alpha)$, we get the total d-cost is $\sum_{S \in \mathcal{C}} |S \cap R| = \|A_{\mathcal{C}}\|_1 \le (12\alpha \log_2 N) \cdot T$.

Application to Shortest-Path Labelings. To use this result for shortest-path labelings, we would again use the same reduction and the same max-distance oracle as in Section 3. We can try all the polynomially many guesses for u and T.

6 Minimizing ℓ_p Norms

We now turn to approximating ℓ_p norms of the element loads for $p \in [1, \log N]$. Specifically, we want a set cover $\mathcal{C}$ with the ℓ_p-norm of the loads $\|A_{\mathcal{C}}\|_p$ only logarithmically larger than $\|A_{\mathcal{C}^*}\|_p$ for any other set cover $\mathcal{C}^*$. (Recall that the vectors $A_{\mathcal{C}}$ and $A_{\mathcal{C}^*}$ are only defined on the relevant elements.) The round-t costs are now $c_{p,t}(e) := (A_t(e)+1)^p - A_t(e)^p$ for $e \in R$; so

$$c_{p,t}(S) \quad := \quad \sum_{e \in S \cap R} ((A_t(e)+1)^p - A_t(e)^p),$$

(the sum being only over the relevant items), and the algorithm picks a set that (approximately) minimizes $\frac{c_{p,t}(S)}{|S \setminus X_t|}$. Again, X_t is the set of elements covered prior to round t. $A_t(e)$ is the load of element e at the beginning of round t. To make the analysis easier, we set $A_1(e) = p$ for all relevant elements $e \in R$. The potential function is now a polynomial: $\Phi_p(t) := \sum_{e \in R} A_t(e)^p$. It immediately follows that if we pick set S_t in round t, $\Phi_p(t+1) - \Phi_p(t) = c_{p,t}(S_t)$. Initially $\Phi_p(1) = |R| \cdot p^p$; we will have to deal with this issue.

Lemma 9. *For $b := (e-1)$ and any t,* $\sum_{S \in \mathcal{C}^*} c_{p,t}(S) \le b \cdot p \cdot \Phi_p(t)^{(p-1)/p} \cdot \|A_{\mathcal{C}^*}\|_p$.

Proof. By the definition of $c_{p,t}(\cdot)$, we know that

$$\sum_{S\in\mathcal{C}^*} c_{p,t}(S) = \sum_{S\in\mathcal{C}^*}\sum_{e\in S\cap R}((A_t(e)+1)^p - A_t(e)^p) \le \sum_{S\in\mathcal{C}^*}\sum_{e\in S\cap R}(\mathrm{e}-1)\,p\,A_t(e)^{p-1},$$

(which follows from Fact 1 below and the observation that $A_t(e) \ge A_1(e) \ge p$)

$$\begin{aligned}
&= (\mathrm{e}-1)\,p\sum_{e\in R}\left[A_t(e)^{p-1}\sum_{S\in\mathcal{C}^*:e\in S} 1\right] = (\mathrm{e}-1)\,p\sum_{e\in R} A_t(e)^{p-1}A_{\mathcal{C}^*}(e)\\
&\le (\mathrm{e}-1)\,p\,\|A_t\|_p^{p-1}\,\|A_{\mathcal{C}^*}\|_p \qquad \text{(by Hölder's inequality)}\\
&\le (\mathrm{e}-1)\,p\cdot\Phi(t)^{(p-1)/p}\cdot\|A_{\mathcal{C}^*}\|_p
\end{aligned}$$

Note that we used the fact that we initialized $A_1(e) \ge p$. □

Fact 1 *For any real $r \ge 1$ and $x \ge r$, $(x+1)^r - x^r \le (e-1)\,r\,x^{r-1}$.*

Proof. Observe that $(x+1)^r - x^r = x^r((1+x^{-1})^r - 1)$, thus

$$x^r\sum_{j=1}^{\infty}\binom{r}{j}x^{-j} \le x^r\sum_{j=1}^{\infty}\frac{r^j}{j!}x^{-j} \le x^r\frac{r}{x}\sum_{j=1}^{\infty}\frac{(r/x)^{j-1}}{j!} \le r\,x^{r-1}\sum_{j=1}^{\infty}\frac{1}{j!} \le (\mathrm{e}{-}1)\,r\,x^{r-1},$$

where we used the inequality $x \ge r$. □

Corollary 1. *At any time t, there exists a set S with $\frac{c_{p,t}(S)}{|S\setminus X_t|} \le \frac{2b\cdot p\cdot\Phi_p(t)^{(p-1)/p}\cdot\|A_{\mathcal{C}^*}\|_p}{|U\setminus X_t|}$. Hence the max-density algorithm picks a set with cost-to-coverage ratio at most α times that.*

Proof. If there exists a set in $\mathcal{C}^*$ that satisfies the above property, we are done. Hence imagine that no set in $\mathcal{C}^*$ satisfies it. Then

$$\sum_{S\in\mathcal{C}^*}\frac{|S\setminus X_t|}{|U\setminus X_t|} \le \sum_{S\in\mathcal{C}^*}\frac{c_{p,t}(S)}{2b\cdot p\cdot\Phi_p(t)^{(p-1)/p}\cdot\|A_{\mathcal{C}^*}\|_p} \le \frac{1}{2},$$

where the last inequality is from Lemma 9. But since all elements in U need to be covered by $\mathcal{C}^*$, this quantity is at least 1, a contradiction. □

Lemma 10. *For $\mathcal{C}$ produced by the algorithm, $\|A_{\mathcal{C}}\|_p \le O(\alpha\log N)\cdot\|A_{\mathcal{C}^*}\|_p$.*

Proof. We define phase $i \in \{0,1,\dots,\log_2 N\}$ to consist of rounds t where $|U\setminus X_t| \in (\frac{N}{2^{i+1}}, \frac{N}{2^i}]$. Let $\beta := 2\,\alpha\,b\,p$. Let t^* denote the last round where $\Phi_p(t^*)^{1/p} \le 4\beta\|A_{\mathcal{C}^*}\|_p$, and let i^* denote the phase containing t^*. Note the starting potential was $\Phi_p(1) = |R|\cdot p^p \le |R|\,\beta^p \le \beta^p\|A_{\mathcal{C}^*}\|_p^p$; hence $t^* \ge 1$.

Consider any phase $i \ge i^*$, and let I (resp., F) denote the values of Φ_p at the start (resp., end) of phase i. By our choice of t^* and hence of i^*, it follows that $F^{1/p} \ge \Phi_p(t^*+1)^{1/p} > 4\beta\|A_{\mathcal{C}^*}\|_p$. Moreover, the cost-to-coverage ratio of sets picked in phase i is at most $\frac{\beta\cdot F^{(p-1)/p}\cdot\|A_{\mathcal{C}^*}\|_p}{N/2^{i+1}}$ (using Corollary 1), and at most

$N/2^i$ elements are covered in this phase. Consequently the total cost incurred during phase i is at most $2\beta \cdot F^{(p-1)/p} \cdot \|A_{\mathcal{C}^*}\|_p$; moreover, this total cost equals the increase in potential, $F - I$. We can rewrite the resulting inequality as:

$$I^{\frac{1}{p}} \geq F^{\frac{1}{p}} \left(1 - \frac{2\beta \|A_{\mathcal{C}^*}\|_p}{F^{\frac{1}{p}}}\right)^{\frac{1}{p}} \geq F^{\frac{1}{p}} \cdot e^{\frac{-4\beta\|A_{\mathcal{C}^*}\|_p}{p \cdot F^{\frac{1}{p}}}} \geq F^{\frac{1}{p}} \cdot \left(1 - \frac{4\beta \|A_{\mathcal{C}^*}\|_p}{p \cdot F^{\frac{1}{p}}}\right).$$

The second inequality above uses $1 - x \geq e^{-2x}$ for $0 \leq x \leq 1/2$; the final inequality is by $e^y \geq 1 + y$ for all y. This gives us that $F^{1/p} - I^{1/p} \leq \frac{4\beta}{p} \cdot \|A_{\mathcal{C}^*}\|_p$ for any phase $i \geq i^*$. Now summing over all such phases $i \geq i^*$ (there are at most $\log_2 N$ of them), we obtain $\Phi_p(\mathsf{final})^{1/p} - \Phi_p(t^*)^{1/p} \leq \frac{4\beta}{p} \cdot \log_2 N \cdot \|A_{\mathcal{C}^*}\|_p$. This uses the fact that round t^* lies in phase i^* and $\Phi_p(\cdot)$ is monotone non-decreasing. Finally, using $\Phi_p(t^*) \leq 4\beta\|A_{\mathcal{C}^*}\|_p$ and that $\beta = 2\alpha\, b\, p$, we have $\Phi_p(\mathsf{final})^{1/p} \leq (\frac{4\beta}{p} \log_2 N + 4\beta) \cdot \|A_{\mathcal{C}^*}\|_p = (4\alpha\, b\, (\log_2 N + p)) \cdot \|A_{\mathcal{C}^*}\|_p$. Since $p \leq \log N$, this completes the proof. □

Note that minimizing ℓ_∞ is within constant factors of minimizing $\ell_{\log N}$, so the result subsumes that of Section 4. Moreover, this algorithm does not require us to enumerate over guesses of the optimum load u.

7 Multiple Norms Simultaneously

The approach of the previous section naturally extends to give solutions that are good with respect to multiple ℓ_p norms; we now show how to handle two norms. Specifically, given $p, q \in [1, \log N]$, we want to find a cover $\mathcal{C}$ with $\|A_\mathcal{C}\|_p \leq O(\alpha \log N)\|A_{\mathcal{C}^*}\|_p$ and $\|A_\mathcal{C}\|_q \leq O(\alpha \log N)\|A_{\mathcal{C}^*}\|_q$, where $\mathcal{C}^*$ is some intended "optimal" cover. We assume we know values P, Q such that $\|A_{\mathcal{C}^*}\|_p \approx P$ and $\|A_{\mathcal{C}^*}\|_q \approx Q$.

We define the round-t costs $c_{p,t}$ and $c_{q,t}$ as in (6), and the potentials $\Phi_p(t)$ and $\Phi_q(t)$ as in (6). We initialize the loads $A_1(e)$ to $\max\{p, q\} \leq \log N$, and run the algorithm where picking the set S minimizing the "combined" ratio:

$$\frac{1}{|S \setminus X_t|} \cdot \left(\frac{c_{p,t}(S)}{p \cdot \Phi_p(t)^{(p-1)/p} \cdot P} + \frac{c_{q,t}(S)}{q \cdot \Phi_q(t)^{(q-1)/q} \cdot Q}\right) \tag{5}$$

Lemma 11. *At any time t, there exists a set S such that for both $r \in \{p, q\}$,*

$$\frac{c_{r,t}(S)}{|S \setminus X_t|} \leq \frac{3b \cdot r \cdot \Phi_r(t)^{(r-1)/r} \cdot \|A_{\mathcal{C}^*}\|_r}{|U \setminus X_t|}.$$

Proof. Suppose each set $S \in \mathcal{C}^*$ fails the inequality corresponding to either p or q or both, and say $\mathcal{C}^*_p, \mathcal{C}^*_q \subseteq \mathcal{C}^*$ denote the corresponding sets. Then

$$\sum_{r \in \{p,q\}} \sum_{S \in \mathcal{C}^*_r} \frac{|S \setminus X_t|}{|U \setminus X_t|} \leq \sum_{r \in \{p,q\}} \sum_{S \in \mathcal{C}^*_r} \frac{c_{r,t}(S)}{3b \cdot r \cdot \Phi_r(t)^{(r-1)/r} \cdot \|A_{\mathcal{C}^*}\|_r} \leq \frac{1}{3} + \frac{1}{3},$$

where the last inequality uses Lemma 9. But since all elements in U are covered by $\mathcal{C}^*$, this should be at least 1, a contradiction. □

Hence at each step t, there exists some set where the combined ratio objective function (5) for the algorithm has value at most $\frac{6b}{|U \setminus X_t|}$. Thus the algorithm will pick set S_t with objective function value at most α times greater, which guarantees a set S_t with $\frac{c_{r,t}(S_t)}{|S \setminus X_t|} \leq \alpha \cdot \frac{6b \cdot r \cdot \Phi_r(t)^{(r-1)/r} \cdot \|A_{\mathcal{C}^*}\|_r}{|U \setminus X_t|}$ for both $r \in \{p, q\}$. Finally, the analysis from Lemma 10 carries over virtually unchanged for both p, q, the only difference being the definition $\beta := 6b\alpha r$ instead of 6.

The above algorithm extends to finding LSC that is within an $O(\alpha k \log N)$ factor of k different targets with respect to k different ℓ_p norms $p_1, p_2, \ldots, p_k$.

7.1 Non-existence of Simultaneous Optimality

In this section we construct a family of graphs for which no labeling can be simultaneously near-optimal for the total label size T^* and the maximum label size A^*. For any labeling with the total size T and the maximum size M, either T is polynomially bigger than T^* or M is polynomially bigger than M^*.

For a parameter k, the (undirected) graph has three sets of vertices, $A = \{a_1, a_2, \ldots, a_k\}$, $B = \{b_1, b_2, \ldots b_{k^2}\}$, and $C = \{c_{ij} \mid i \in [k^2], j \in [k]\}$, of size k, k^2, and k^3, respectively. Every vertex in A is connected to all vertices in B. Vertices in C are partitioned into k^2 groups of size k each. For every i, the partition $C_i = \{c_{ij} \mid j \in [k]\}$ corresponds to the vertex $b_i \in B$. Each vertex in B is connected to all k vertices in its group in C. There are no other edges in the graph. All edges have length 1, except for the edges from a_1 to B, which are of length $1 - \varepsilon$. The total number of vertices of the graph is $n = \Theta(k^3)$.

Observe that the shortest paths in the graph are as follows:

- For vertices $a, a' \in A$, and every $b \in B$, the path a, b, a' is a shortest a-a' path. For any vertex $a \in A, b \in B$, the edge (a, b) is a shortest path. For any $a \in A$ and $c \in C_i$, the path a, b_i, c is the unique shortest path.
- For vertices $b, b' \in B$, the path b, a_1, b' is the unique shortest path. For any i, the unique shortest path between vertex $b_i \in B$ and any $c \in C_i$ is the edge (b_i, c); for any $c \in C_{i'} \neq C_i$, the path $b_i, a_1, b_{i'}, c$ is the unique shortest path.
- For vertices c, c' in the same group C_i, the path c, b_i, c' is the unique shortest path. For $c \in C_i$ and $c' \in C_j \neq C_i$, the path c, b_i, a_1, b_j, c' is the unique shortest path.

An $O(k^3)$-size labeling is as follows. Each vertex is in its own label. In addition, every vertex in A has all vertices in B in its label. Every vertex in B contains a_1 in its label. Every vertex $c \in C$ has a_1 in its label; moreover, if c belongs to group C_i it has the corresponding vertex $b_i \in B$ its label. The total label size is $k \cdot (k^2 + 1) + 2 \cdot k^2 + 3 \cdot k^3 = O(k^3)$. In this labeling, the vertices in A have labels of size $k^2 + 1$.

There is a different labeling with the maximum label size $O(k)$. Each vertex is in its own label. In addition, every vertex in the graph has all vertices of A in its label. Moreover, if c belongs to group C_i it has the corresponding vertex $b_i \in B$ its label. The total size of this labeling is $\Omega(k^4)$.

Now consider a labeling L with the total size T and the maximum size M. For a vertex $a \in A$, consider shortest paths to all vertices in C. The number of vertices $c \in C$ for which an a–c shortest path contains a vertex of $L(a)$ different from a is at most kM. Therefore labels of $k^3 - kM$ vertices in C must contain a. Thus $T \geq k(k^3 - kM)$, or $T + k^2 M \geq k^4$. Hence, if $T = o(k^4)$, then $M = \Omega(k^2)$, and if $M = o(k^2)$, then $T = \Omega(k^4)$. Therefore T or M is a factor $\Omega(n^{1/3})$ away from the corresponding optima.

Acknowledgments We thank Amos Fiat and Haim Kaplan for interesting discussions. A. Gupta thanks Microsoft Research Silicon Valley, IBM T.J. Watson Research Center, and the IEOR Department at Columbia University for their generous hospitality.

References

1. I. Abraham, D. Delling, A. V. Goldberg, and R. F. Werneck. A Hub-Based Labeling Algorithm for Shortest Paths on Road Networks. In *Proc. SEA'11*, LNCS vol. 6630, pages 230–241, 2011.
2. I. Abraham, D. Delling, A. V. Goldberg, and R. F. Werneck. Hierarchical Hub Labelings for Shortest Paths. In *Proc. ESA'12*, LNCS vol. 7501, pages 24–35, 2012.
3. V.S.A. Kumar, M. V. Marathe, S. Parthasarathy, and A. Srinivasan. A unified approach to scheduling on unrelated parallel machines. *JACM*, 56(5):28–31, 2009.
4. J. Aspnes, Y. Azar, A. Fiat, S. Plotkin, and O. Waarts. On-line routing of virtual circuits with applications to load balancing and machine scheduling. *J. ACM*, 44(3):486–504, 1997.
5. B. Awerbuch, Y. Azar, E. F. Grove, M.-Y. Kao, P. Krishnan, and J. S. Vitter. Load balancing in the $\mathrm{l_p}$ norm. In *FOCS*, pages 383–391, 1995.
6. Y. Azar and A. Epstein. Convex programming for scheduling unrelated parallel machines. In *STOC'05*, pages 331–337, 2005.
7. Y. Azar, L. Epstein, Y. Richter, and G. J. Woeginger. All-norm approximation algorithms. *J. Algorithms*, 52(2):120–133, 2004.
8. V. Chvátal. A Greedy Heuristic for the Set-Covering Problem. *Mathematics of Operations Research*, 4(3):233–235, 1979.
9. E. Cohen, E. Halperin, H. Kaplan, and U. Zwick. Reachability and Distance Queries via 2-hop Labels. *SIAM J. Comput.*, 32, 2003.
10. D. Delling, P. Sanders, D. Schultes, and D. Wagner. Engineering Route Planning Algorithms. In *Algorithmics of Large and Complex Networks*, LNCS vol. 5515, pages 117–139, 2009.
11. U. Feige. A threshold of $\ln n$ for approximating set cover. *J. ACM*, 45(4):634–652, 1998.
12. G. Gallo, M. D. Grigoriadis, and R. E. Tarjan. A Fast Parametric Maximum Flow Algorithm and Applications. *SIAM J. Comput.*, 18:30–55, 1989.
13. C. Gavoille, D. Peleg, S. Pérennes, and R. Raz. Distance Labeling in Graphs. *Journal of Algorithms*, 53:85–112, 2004.
14. A. V. Goldberg. A Practical Shortest Path Algorithm with Linear Expected Time. *SIAM Journal on Computing*, 37:1637–1655, 2008.
15. G. Kortsarz and D. Peleg. Generating Sparse 2-Spanners. *J.Alg.*, 17:222–236, 1994.
16. M. Thorup. Undirected Single-Source Shortest Paths with Positive Integer Weights in Linear Time. *JACM*, 46:362–394, 1999.