A general approach for incremental approximation and hierarchical clustering

Guolong Lin* Chandrashekhar Nagarajan† Rajmohan Rajaraman‡
David P. Williamson§

Abstract

We present a general framework and algorithmic approach for incremental approximation algorithms. The framework handles cardinality constrained minimization problems, such as the k-median and k-MST problems. Given some notion of ordering on solutions of different cardinalities k, we give solutions for all values of k such that the solutions respect the ordering and such that for any k, our solution is close in value to the value of an optimal solution of cardinality k. For instance, for the k-median problem, the notion of ordering is set inclusion and our incremental algorithm produces solutions such that any k and k', $k < k'$, our solution of size k is a subset of our solution of size k'. We show that our framework applies to this incremental version of the k-median problem (introduced by Mettu and Plaxton [30]), and incremental versions of the k-MST problem, k-vertex cover problem, k-set cover problem, as well as the uncapacitated facility location problem (which is not cardinality-constrained). For these problems we either get new incremental algorithms, or improvements over what was previously known. We also show that the framework applies to hierarchical clustering problems. In particular, we give an improved algorithm for a hierarchical version of the k-median problem introduced by Plaxton [31].

1 Introduction

1.1 Incremental problems A company is building facilities in order to supply its customers. Because of limited capital, it can only build a few at this time, but intends to expand in the future in order to improve its customer service. Its plan for expansion is a sequence of facilities that it will build in order as it has funds. Can it plan its future expansion in such a way that if it opens the first k facilities in its sequence, this solution is close in value to that of an optimal solution that opens any choice of k facilities? The company's problem is the *incremental* k-median problem, and was originally proposed by Mettu and Plaxton [30][1]. The standard k-median problem has been the object of intense study in the algorithms community in the past few years. Given the locations of a set of facilities and a set of clients in a metric space, a demand for each client, and a parameter k, the *k-median problem* asks to find a set of k facilities to *open* such that the sum of the demand-weighted distances of the clients to the nearest open facility is minimized. In the incremental k-median problem, we are given the input of the k-median problem without the parameter k and must produce a sequence of the facilities. For each k, consider the ratio of the cost of opening the first k facilities in the ordering to the cost of an optimal k-median solution. The goal of the problem is to find an ordering that minimizes the maximum of this ratio over all values of k. An algorithm for the problem is said to be *α-competitive* if the maximum of the ratio over all k is no more than α. This value α is called the *competitive ratio* of the algorithm. We will also consider randomized algorithms for the incremental k-median problem. For a randomized algorithm, we consider the ratio of the expected cost of opening the first k facilities in the ordering to the cost of an optimal k-median solution. The algorithm is α-competitive if this ratio is at most α for all k.

In a similar manner, one can also define natural incremental versions of any cardinality constrained minimization problems, including the k-minimum spanning tree problem (k-MST), k-vertex cover, and k-set cover problems. In the standard weighted vertex cover problem, we are given an undirected graph with weights on the vertices and we wish to find a minimum-weight sub-

*College of Computer and Information Science, Northeastern University, Boston, MA 02115. Email: lingl@ccs.neu.edu.

†School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY 14853. Email: chandra@orie.cornell.edu.

‡College of Computer and Information Science, Northeastern University, Boston, MA 02115. Email: rraj@ccs.neu.edu.

§School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY 14853. Email: dpw@cs.cornell.edu. Supported in part by NSF grant CCF-0514628 and an IBM Faculty Partnership Award.

[1] Mettu and Plaxton call it the *online* median problem, but we would like to draw a distinction between incremental and online problems.

set of vertices S such that every edge has at least one endpoint in S. In the k-vertex cover problem, we wish to find a minimum-weight set of vertices that covers at least k edges. In the incremental k-vertex cover problem, we wish to find a sequence of vertices, such that if we choose the smallest prefix of vertices in the sequence that covers at least k edges, this solution is close in value to that of the optimal k-vertex cover solution. An incremental version of the facility location problem, which is not cardinality-constrained, has also been defined [31].

Perhaps less obviously, many hierarchical clustering problems can also be cast as incremental problems. In hierarchical clustering, we give clusterings with k clusters for all values of k by starting with n clusters and repeatedly merging selected pairs of clusters until all points are in a single cluster. Given some objective function on a k-clustering, again we would like to ensure that for any k, the cost of our k-clustering obtained in this way is not too far away from the cost of an optimal k-clustering. The connection with incremental problems is this: for the incremental k-median problem, we insist that for any k, k', with $k < k'$, our solution with k facilities is *ordered* with respect to our solution on k' facilities; namely, the smaller solution is a subset of the larger. In hierarchical clustering, for any k, k', with $k < k'$, our k-clustering must also be ordered with respect to our k'-clustering; namely, the k'-clustering must be a refinement of the k-clustering. We can then consider various clustering criteria: minimize the maximum radius from a cluster center (k-center), minimize the sum of demand-weighted distances of points to their cluster center (k-median), or minimize the sum of demand-weighted distances-squared of points to their cluster center (k-means). From these we obtain hierarchical variants, which we say are α-competitive if for any k, the k-clustering produced by our algorithm is at most α times the cost of an optimal k-clustering under the given objective.

1.2 Our contribution

Our central contribution is to give a general approach for solving incremental optimization problems. We then apply this to the incremental versions of the k-median, k-means, k-MST, facility location, k-vertex cover, and k-set cover problems. Furthermore, we apply it to hierarchical clustering problems with the k-median and k-means objective functions. We state our approach in terms of *posets* on solutions to the problems, in which two solutions are comparable in the poset if they obey the ordering imposed by the incremental solution (e.g. if one of the k-median solutions is a subset of the other, or one of the k-vertex cover solutions is a subset of the other, or one of the k-clusterings is a refinement of the other). Each solution in the poset has a cost, as defined by the underlying optimization problem. In addition, we associate a benefit with each solution that models the constraint of the optimization problem (corresponding, for example, to the number of unopened facilities, or the number of edges covered). The goal of the incremental problem is to find a chain of solutions such that for any b, the least element in the chain (according to the partial order) that has benefit at least b has cost close to that of an optimal solution of benefit at least b.

To obtain a competitive solution for a given incremental problem in polynomial-time, our algorithm relies on an α-approximation algorithm for the underlying offline optimization problem. It also relies on an *augmentation* subroutine that, given two solutions of benefits b, b', $b < b'$, which are incomparable in the poset, finds another solution of benefit at least b' that is comparable in the poset to the solution of benefit b. If one can show that this solution has cost no more than a linear combination of the costs of the original two solutions, then one can obtain an $O(\alpha)$-competitive algorithm, where the constant in the big-O depends on the constants in the linear combination. The basic idea of the incremental algorithm is to build a chain of solutions of geometrically increasing cost by repeatedly applying the augmentation subroutine to the current solution in the chain and a solution generated by the offline approximation algorithm that has cost no more than the next bound in the geometrically increasing order. Similar ideas are implicit in the minimum latency approximation algorithm of Blum et al [4], the incremental facility location algorithm of Plaxton [31] and the hierarchical k-center algorithm of Dasgupta and Long [12]. Choosing a random shift of the buckets as in Goemans and Kleinberg [17] and Dasgupta and Long [12] gives improved randomized algorithms.

In some cases, we are able to improve the competitive ratio still further. In particular, if there exists a Lagrangean multiplier preserving ρ-approximation algorithm for the problem in which Lagrangean relaxation has been applied to the benefit constraint, we are able to give the same result as above in which this algorithm replaces the α-approximation algorithm. This yields improved competitive ratios in the cases where we have such algorithms with performance guarantees ρ better than the best known performance guarantee α for the problem with the benefit constraint. In particular, there is a Lagrangean multiplier preserving 2-approximation algorithm for the facility location problem (due to Jain et al. [25]), which we can use in place of a $(3+\epsilon)$-approximation algorithm for the k-median problem (due to Arya et al. [3]), yielding improvements in the competitive ratios for the incremental k-median problem, hierarchical k-median problem, and hierarchical k-means problem.

We summarize our main results in Table 1.

1.3 Related work All of the optimization problems studied in this paper are NP-hard and have been extensively studied with respect to their approximability. Several approximation algorithms are known for the vertex cover and set cover problems (see, for example, [23, 34]). Our incremental k-vertex cover algorithm relies on a 2-approximation algorithm for k-vertex cover, while our incremental k-set cover algorithm relies on an $O(\log n)$-approximation algorithm for k-set cover [6, 15, 22, 29, 32].

The k-median problem and the related uncapacitated facility location problems have been the objects of intense study in the algorithms community in the past few years. The currently best known approximation algorithms for these problems have performance guarantees of $3 + \epsilon$ (due to Arya et al. [3]) and 1.52 (due to Mahdian, Ye, and Zhang [28]) respectively. Of interest to us is the best known Lagrangean multiplier preserving approximation algorithm for the facility location problem with performance guarantee of 2, which is due to Jain et al. [25]. The best currently known approximation algorithm for the k-MST problem has a performance guarantee of 2 (Garg [16]).

There has been a lot of previous work on incremental approximation algorithms, but it was usually done on a problem-by-problem basis. Mettu and Plaxton [30] introduce the incremental k-median problem, and give a 29.86-competitive algorithm for it. Their algorithm runs in near linear time and their argument also applies when the distances satisfy a weaker version of triangle inequality, yielding an $O(1)$-competitive solution for the incremental k-means problem. Plaxton [31] introduces the incremental facility location problem and gives an $(4 + \epsilon)\alpha$-competitive algorithm for it, given any α-approximation algorithm for the uncapacitated facility location problem, resulting in a 12.16-competitive algorithm. González [18] gives a 2-approximation algorithm for the k-center problem, which is also a 2-competitive algorithm for the incremental k-center problem. Implicit in the work of Charikar et al. [7] on incremental clustering are a deterministic 8-competitive algorithm and a randomized $2e$-competitive algorithm for the hierarchical k-center problem. Dasgupta and Long [12] explicitly introduce the idea of finding competitive hierarchical clusterings, and derive the same bounds as above for the hierarchical k-center problem. Plaxton [31] gives an 8α-competitive algorithm for the hierarchical k-median problem, given an α-competitive algorithm for the incremental k-median problem. Using the algorithm of Mettu and Plaxton [30] gives a 238.88-competitive algorithm for the hierarchical k-median problem. The work of [31] also includes an $O(1)$-competitive algorithm for the hierarchical k-means problem. Implicit in work on the minimum latency problem is a number of different algorithms for the incremental k-MST problem; given an α-approximation algorithm for the k-MST problem, the work of Blum et al. [4] yields an 4α-competitive algorithm, while a randomized $e\alpha$-competitive algorithm is implicit in Goemans and Kleinberg [17]. Other work on incremental approximation algorithms includes incremental flow (Hartline and Sharp [21]) and incremental bin packing (Codenotti et al. [11]).

Independently Chrobak, Kenyon, Noga, and Young [9] also discovered the same $(24 + \epsilon)$-competitive deterministic and $(6e + \epsilon)$-competitive randomized algorithms as ours for the incremental k-median problem. They also consider the incremental version of a median problem in which the goal is to minimize the number of medians required to satisfy a given cost constraint. These results are derived by a reduction from a new problem, which they call online bribery, for which tight upper and lower bounds are established. Chrobak et al. [9] also extend their work to fractional k-medians, approximately metric distance functions, which include the k-means objective, and bicriteria approximations.

As a paradigm for dealing with uncertainty, incremental approximations are most closely related to online algorithms, stochastic optimization, and universal approximations. The study of online algorithms considers problems in which the input is revealed over time, and the algorithm must make decisions without knowledge of future inputs [5, 14, 33]. The study of stochastic optimization (e.g., see [13, 19, 20, 24]) considers problems in which the cost of future decisions may be significantly different than those now: the future is unknown, but is chosen randomly from one of a number of different possible scenarios which are known (possibly given as a black box). In contrast, an incremental algorithm for a problem performs all of the computation offline and outputs a single chain of solutions such that for every possible benefit constraint there is a valid solution that is close to the optimal. Our measure of performance for incremental solutions is modeled on the measure of competitive ratio from online algorithms. The notion of universal approximations studied in [27] considers a much stronger notion of uncertainty in the sense that the number of possibilities for the unknown portion of the input is exponential in the size of the problem. As a result, the competitive ratios achievable within the incremental framework are much smaller than the approximations achievable within the universal framework.

Problem	Prev known	Competitive ratio					
		Via optimal		Via approx		Via LMP approx	
		Det	Rand	Det	Rand	Det	Rand
Incremental k-median	29.86 [30]	8^*	$2e^*$	$24 + \epsilon^*$	$6e + \epsilon^*$	16	$4e$
Incremental k-MST	8 [4], $2e$ [17]	4	e	8	$2e$		
Incremental k-vertex cover		4	e	8	$2e$		
Incremental k-set cover		4	e	$O(\log n)$	$O(\log n)$		
Incremental facility location	12.16 [31]	4	e	12.16	$1.52e$		
Hierarchical k-median	238.88 [31]	20.71	10.03	$62.13 + \epsilon$	$30.09 + \epsilon$	41.42	20.06

Table 1: Our summary of results. The first column gives the best previously known competitive ratio for a polynomial-time algorithm. The second and third state the competitive ratio for incremental solutions obtained using optimal algorithms for the benefit-constrained problems and are thus non-polynomial-time algorithms; they should be viewed as existential results. The fourth and fifth state the competitive ratio for our polynomial-time algorithms via an α-approximation algorithm. The sixth and seventh give the competitive ratio for our polynomial-time algorithms via a Lagrangean multiplier preserving ρ-approximation algorithm. The results with * were independently obtained by Chrobak, Kenyon, Noga, and Young [9].

2 A general framework for incremental optimization

In this section, we present a general framework for incremental optimization (Section 2.1) and a generic approximation algorithm for incremental optimization problems that lie within this framework (Section 2.2).

2.1 Problem definitions The problems we consider in this paper are all minimization problems and share the following characteristics. Each optimization problem Π can be specified by a quadruple $\langle U, \mathsf{ben}, \mathsf{cost}, p\rangle$, where U is a set of feasible solutions, $\mathsf{ben} : U \to \mathcal{R}$ and $\mathsf{cost} : U \to \mathcal{R}$ are *benefit* and *cost* functions, respectively, and the goal is to seek a solution S that minimizes $\mathsf{cost}(S)$ subject to the condition $\mathsf{ben}(S) \geq p$. We refer to Π as an offline problem to distinguish it from its incremental version, which we now define. We introduce a binary relation $\preceq$, which induces a partial order on U, i.e., $\langle U, \preceq\rangle$ is a *poset*. Throughout this paper, we focus on benefit and cost functions that are monotonically non-decreasing with respect to the partial order; that is, if $S \preceq S'$, then $\mathsf{ben}(S) \leq \mathsf{ben}(S')$ and $\mathsf{cost}(S) \leq \mathsf{cost}(S')$. The incremental version of Π is specified by the quadruple $\langle U, \preceq, \mathsf{ben}, \mathsf{cost}\rangle$ and seeks a *chain* $\mathcal{C}$ of $\langle U, \preceq\rangle$. Define the competitive ratio of $\mathcal{C}$ as

$$\sup_{0 \leq p \leq \mathsf{ben}(U)} \frac{\mathsf{cost}(\pi(\mathcal{C}, p))}{\mathsf{cost}(\mathsf{Opt}(p))},$$

where $\pi(\mathcal{C}, p)$ denotes the smallest indexed element of $\mathcal{C}$ whose benefit is at least p and $\mathsf{cost}(\mathsf{Opt}(p))$ is the cost of an optimal solution for the offline problem for benefit p, namely $\langle U, \mathsf{ben}, \mathsf{cost}, p\rangle$.

2.2 A generic incremental approximation algorithm The core of each of our approximation algorithms for incremental optimization problems is a subroutine for augmenting a given solution to achieve a certain benefit. In this section, we present a sufficient condition for the existence of such an augmentation. By repeatedly invoking this augmentation subroutine (which is specific to the particular problem), we show how to derive a sequence that has a good competitive ratio. We begin by defining the augmentation property. For convenience, let B_{max} denote $\max_{S \in U} \mathsf{ben}(S)$, the maximum benefit achieved by a feasible solution.

DEFINITION 2.1. (γ, δ)*-Augmentation: For every solution S of U and every real $p \leq B_{max}$, there exist an augmentation S' and reals $\gamma, \delta \geq 0$ such that*

1. $S \preceq S'$.
2. $\mathsf{cost}(S') \leq \gamma \mathsf{cost}(S) + \delta \mathsf{cost}(\mathsf{Opt}(p))$.
3. $\mathsf{ben}(S') \geq p$.

Let $\mathsf{Augment}(S, p, \gamma, \delta)$ *denote a subroutine that computes such an augmentation.*

We now present two generic incremental optimization algorithms, given an augmentation subroutine. One is deterministic, while the other is randomized. Since these two algorithms share the same structure, differing only in the parameter setting (the Initialization step below), they are shown together. In the subsequent sections, we show that for each of the problems we consider in this paper, the augmentation subroutine can be implemented using an approximation algorithm

Algorithm 1 $\textsc{IncApprox}(\gamma, \delta)$

1. Initialization:

   1D: (Deterministic) $i = 0$, $S_0 = \emptyset$, $\beta = 2\gamma$, $\beta_0 = 1$.

   1R: (Randomized) $i = 0$, $S_0 = \emptyset$, β is the minimizer of $\frac{\beta-1}{(1-\gamma/\beta)\ln\beta}$, $\beta_0 = \beta^X$, where X is uniform from $[0, 1)$.

2. Iteration i: $S_{i+1} = \mathsf{Augment}(S_i, p, \gamma, \delta)$, where p is the largest value for which $\mathsf{cost}(\mathsf{Augment}(S_i, p, \gamma, \delta))$ is at most $\beta_0\beta^{i+1}$ (e.g., do a binary search on p).
3. Termination: If $\mathsf{ben}(S_i) \neq B_{max}$, $i \leftarrow i + 1$, go to step 2; Otherwise, return sequence $S_1, \cdots, S_i$.

to the offline optimization problem for suitable choices of γ and δ.

Remark 2.1. *For some applications discussed in this paper, most notably the incremental and hierarchical median problems, the poset induced by the partial order is, in fact, a ranked poset; that is, every maximal chain in the poset is of the same length. For these problems, we can replace the chain $\mathcal{C}$ that is output by the above incremental algorithm by any maximal chain that contains $\mathcal{C}$, without increasing the competitive ratio.*

Theorem 2.1. *If (γ, δ)-augmentation holds for reals $\gamma \geq 1$, $\delta > 0$, then (i) $\textsc{IncApprox}(\gamma, \delta)$ (Deterministic) computes an incremental solution with competitive ratio $4\gamma\delta$; (ii) $\textsc{IncApprox}(\gamma, \delta)$ (Randomized) computes an incremental solution with competitive ratio $\min_\beta \frac{\delta(\beta-1)}{(1-\gamma/\beta)\ln\beta}$, which equals $e\delta$, when $\gamma = 1$.*

Proof. Fix a real $p \leq B$. Let S^* denote an optimal solution for the instance with benefit p. Let i be the smallest integer such that $\frac{\delta \mathsf{cost}(S^*)}{1-\gamma/\beta} \leq \beta_0\beta^i$. It follows that $\mathsf{cost}(S^*) > (\beta_0\beta^i/\beta) \cdot (1 - \gamma/\beta)/\delta$.

By the augmentation property, we have $\mathsf{ben}(S_i) \geq p$ since $\mathsf{cost}(S_{i-1}) \leq \beta_0\beta^{i-1}$ and $\gamma\beta_0\beta^{i-1} + \delta\mathsf{cost}(S^*) \leq \beta_0\beta^i$. Note that for the examples treated in this paper, $\mathsf{cost}(S_0)$ is either 0 or could be scaled to 1. Now we analyze the two versions of the algorithm.

Deterministic case: We lower bound $\mathsf{cost}(S^*)$ by $\frac{\beta_0\beta^i}{\beta} \cdot \frac{1-\gamma/\beta}{\delta}$, and obtain the following upper bound on the competitive ratio of $\mathcal{C}$.

$$\begin{aligned} & \mathsf{cost}(\pi(\mathcal{C}, p))/\mathsf{cost}(\mathsf{Opt}(p)) \\ \leq\ & \mathsf{cost}(S_i)/\mathsf{cost}(\mathsf{Opt}(p)) \\ \leq\ & \frac{\beta_0\beta^i}{\frac{\beta_0\beta^i}{\beta} \cdot \frac{1-\gamma/\beta}{\delta}} = \beta^2\delta/(\beta - \gamma). \end{aligned}$$

The above bound is minimized when $\beta = 2\gamma$, thus yielding a $4\delta\gamma$ competitive ratio.

Randomized case: Since β_0 is a random variable β^X, where X is uniform in $[0, 1)$, it follows that $\frac{\beta_0\beta^i}{\frac{\delta\mathsf{cost}(S^*)}{1-\gamma/\beta}}$ is a random variable β^Y, where Y is uniform in $[0, 1)$. Thus the expectation of the ratio $\frac{\beta_0\beta^i}{\frac{\delta\mathsf{cost}(S^*)}{1-\gamma/\beta}}$ is $\int_0^1 \beta^y dy = \frac{\beta-1}{\ln\beta}$. We conclude that the competitive ratio is

$$\begin{aligned} E\left[\frac{\beta_0\beta^i}{\mathsf{cost}(S^*)}\right] &= E\left[\frac{\beta_0\beta^i}{\frac{\delta\mathsf{cost}(S^*)}{1-\gamma/\beta}}\right] \cdot \frac{\delta}{1-\gamma/\beta} \\ &= \frac{\delta(\beta - 1)}{(1 - \gamma/\beta)\ln\beta}. \end{aligned}$$

We select β to minimize the last term. In particular, with $\gamma = 1$, we set $\beta = e$, obtaining a ratio of $e\delta$. □

3 Applications

In this section, we apply our framework of Section 2 to incremental versions of several classical optimization problems. Due to space constraints, we have omitted several proofs from this extended abstract.

3.1 The incremental k-MST problem

Given a complete graph $G = (V, E)$, $|V| = n$, with metric cost function $w : E \to Q^+$ and a special $r \in V$, the (rooted) *k-MST problem* seeks a minimum-cost subgraph of G that spans at least k vertices, including r. In the incremental k-MST problem, we seek a sequence of $n-1$ edges of E, $e_1, e_2, \ldots, e_{n-1}$ such that for any $k \in [2, n]$, the first $k - 1$ edges of the sequence span k vertices including r. For each k, consider the ratio of the sum of the cost of the first $k-1$ edges to the cost of an optimal k-MST of G that covers r. The goal of incremental k-MST is to seek a sequence of edges that minimizes the maximum of this ratio, over all k.

In our framework, U is the set of all connected subgraphs of G that contain r, $\preceq$ is the $\subseteq$ relation of the edge subsets. The benefit of a solution is the number of vertices it spans, and the cost is the sum of the edge weights.

Lemma 3.1. *There exists a $(1, 1)$-augmentation for the k-MST problem, and a $(1, \alpha)$-augmentation that can be implemented in poly-time, where $\alpha = 2$.* □

Theorem 3.1. *There exists a solution to incremental k-MST problem with competitive ratio 4. A deterministic solution with competitive ratio 4α and a randomized solution with competitive ratio $e\alpha$ can be computed efficiently, where $\alpha = 2$.*

Proof. Immediate from Lemma 3.1, Theorem 2.1, and the 2-approximation for k-MST due to Garg [16]. □

We note that this computation of 8-competitive incremental MST is implicit in the work of Blum et al [4].

3.2 Incremental and hierarchical median problems

3.2.1 The incremental k-median problem Given the locations of a set F of $|F| = n_f$ facilities and a set C of $|C| = n_c$ clients in a metric space, *the k-median problem* asks to find a set of k facilities to *open* such that the sum of the demand-weighted distances of the clients to the nearest open facility is minimized. Let c_{ij} denote the distance between any two locations i and j. In the incremental k-median problem, we seek an ordering of the facilities. For each k, consider the ratio of the cost of opening the first k facilities in the ordering to the cost of an optimal k-median solution. The goal is to find an ordering that minimizes the maximum of this ratio over $k = 1, \ldots, n_f$.

We model the incremental median problem using our framework of Section 2 by the quadruple $\langle U, \preceq, \mathsf{ben}, \mathsf{cost}\rangle$. The set $U = 2^F$ is the set of all feasible solutions, each solution represented by the set of open facilities. The binary relation is given as $S_1 \preceq S_2$ iff $S_1 \supseteq S_2$, $\mathsf{ben}(S)$ equals $n_f - |S|$, and $\mathsf{cost}(S)$ is the cost of connecting the clients to their nearest facilities in S. The output of our incremental approximation algorithm is a chain of subsets of the facilities, where each chain element (subset of facilities) is a subset of the previous element. As shown in Theorem 3.2 below, the desired sequence of facilities for the incremental median problem is simply a concatenation of the differences between consecutive sets of this chain, presented in reverse order. The main claim of the following lemma is implicit in [26] and [10].

LEMMA 3.2. *There exists a $(1, 2)$-augmentation for the incremental median problem. A $(1, 2\alpha)$-augmentation can be efficiently implemented, where $\alpha = 3 + \epsilon$.*

Proof. Let S_2 ($\mathsf{ben}(S_2) < p$) be a set of facilities. We would like to augment it to get a benefit of at least p. Let S_1 be a set of facilities with benefit p. According to the definition, $|S_2| > |S_1|$. We aim to find a subset S such that $S_2 \preceq S$, i.e., $S \subset S_2$, and $|S| \leq |S_1|$.

For any location (client or facility) j, let $d_1(j)$ (resp., $d_2(j)$) be the closest facility to j in S_1 (resp., S_2). For any client j let us bound the distance $c_{j,d_2(d_1(j))}$.

$$\begin{aligned} c_{j,d_2(d_1(j))} &\leq c_{j,d_1(j)} + c_{d_1(j),d_2(d_1(j))} \\ &\leq c_{j,d_1(j)} + c_{d_1(j),d_2(j)} \\ &\leq c_{j,d_1(j)} + c_{j,d_1(j)} + c_{j,d_2(j)} \\ &= 2c_{j,d_1(j)} + c_{j,d_2(j)}, \end{aligned}$$

where the second inequality follows since $d_2(d_1(j))$ is the closest median in S_2 to $d_1(j)$. Define $S = \{d_2(i) : i \in S_1\}$; that is, S is the set of facilities in S_2 that are closest to the facilities in S_1. Let $d(j)$ be the closest facility in S for a location j. For any client j, $c_{j,d(j)} \leq c_{j,d(d_1(j))} = c_{j,d_2(d_1(j))} \leq 2c_{j,d_1(j)} + c_{j,d_2(j)}$. Multiplying by the client demand and summing over all clients, we obtain $\mathsf{cost}(S) \leq \mathsf{cost}(S_2) + 2\mathsf{cost}(S_1)$. Note that $S \subset S_2$ and $|S| \leq |S_1|$.

Using an optimum solution (resp., α-approximate solution [3]) to the k-median problem for S_1 proves the first (resp., second) assertion. □

THEOREM 3.2. *There exists a solution to the incremental median problem with competitive ratio 8. A deterministic solution with competitive ratio 8α and a randomized solution with competitive ratio $2e\alpha$ can be computed efficiently, where $\alpha = 3 + \epsilon$.*

Proof. The existence and computability of chains of $\langle U, \preceq\rangle$ with the desired competitive ratios follow immediately from Lemma 3.2 and Theorem 2.1. To convert a given chain $\mathcal{C}$ of facility sets into a sequence of medians, we simply generate a maximal chain containing $\mathcal{C}$ and concatenate the differences between consecutive sets of this chain in reverse order. By the definition of competitive ratio (see Section 2.1), the competitive ratio of the chain is at least that of the median sequence, thus completing the proof of the theorem. □

3.2.2 The hierarchical k-median problem We define *an assignment* of a k-median solution as function from clients to facilities that assigns each client to an open facility in the solution. In the hierarchical k-median problem, we give an ordering of facilities along with assignments $a_1, \ldots, a_{n_f}$ such that the assignment a_k assigns clients only to the first k facilities in the ordering; this corresponds to a clustering with k clusters. To ensure that the clusterings are formed by merging pairs of clusters, we require that for any two assignments a_k and a_{k-1} that a_{k-1} can be obtained from a_k by reassigning all the clients assigned to the kth facility in the ordering to a single facility earlier in the ordering. Now consider the ratio of the cost of assignment a_k to the cost of an optimal k-median solution. The goal of the problem is to find an ordering of facilities and a valid sequence of assignments so as to minimize the maximum of this ratio over all $k = 1, \ldots, n_f$.

We show how to cast the hierarchical median problem into our incremental optimization framework. A solution to the k-median problem is represented as a pair (S, a) containing a subset S of facilities and an assignment a of clients to facilities in S. In the incremental k-median problem, the assignment function assigns each client to its nearest available facility. This cannot be assumed for the hierarchical median problem. For any i in S, let $a^{-1}(i)$ be the set of clients assigned to i by a.

Given a solution (S, a), we say that a is *locally-optimal* for S if for all i in S, assigning all clients in $a^{-1}(i)$ to any other single facility in S will not decrease the total cost. We adopt the convention that if the assigment is omitted, the default assignment is to assign each client to its nearest available facility.

In the quadruple $\langle U, \preceq, \mathsf{ben}, \mathsf{cost}\rangle$, we let U be the set of all pairs (S, a) such that a is locally-optimal in (S, a). It is easy to see that U includes all optimal k-median solutions, for all values of k.

DEFINITION 3.1. *We say two solution pairs (S_1, a_1) and (S_2, a_2) in U are* nested *if*

1. $S_1 \subset S_2$;
2. $\forall j \in C$, *if* $a_2(j) \in S_1$ *then* $a_1(j) = a_2(j)$;
3. $\forall j, k \in C$, *if* $a_2(j) = a_2(k)$ *then* $a_1(j) = a_1(k)$.

We denote nested solutions by $(S_1, a_1) \subset (S_2, a_2)$.

We define $\preceq$ as $(S_2, a_2) \preceq (S_1, a_1)$ iff $(S_1, a_1) \subset (S_2, a_2)$, the benefit of a solution (S, a) as $n_f - |S|$, and the cost of (S, a) to be the service cost for the clients according to the assignment a. By definition, the benefit function is monotonically non-decreasing with the partial order. The same holds for the cost function since the assignment in any solution is locally optimal. We now develop an incremental approximation for $\langle U, \preceq, \mathsf{ben}, \mathsf{cost}\rangle$ and show that a chain output by this algorithm can be transformed to a hierarchical ordering of solution pairs, with the desired competitive ratio.

We first prove the following lemma which will be useful in deriving the augmentation lemma.

LEMMA 3.3. *Given a set V_1 of facilities and a solution $(V_2, a_2) \in U$, we can obtain a solution $(V_1, a_1) \in U$ such that (V_1, a_1) and (V_2, a_2) are nested and* $\mathsf{cost}(V_1, a_1) \le 2\mathsf{cost}(V_2, a_2) + \mathsf{cost}(V_1)$.

Proof. Let $d_1(j)$ denote the nearest median in V_1 to the client j. We define two functions P and Q. The function P maps the facilities in $V_2 \setminus V_1$ to their nearest facilities in V_1. This is the "parent" function of the hierarchical algorithms of Dasgupta and Long [12] and Plaxton [31]. The function Q maps any facility i in $V_2 \setminus V_1$ to a facility i_1 in V_1, which services the clients in $a_2^{-1}(i)$ at the least total cost, among all facilities in V_1.

We now create assignment a_1: for any client j, $a_1(j) = a_2(j)$ if $a_2(j) \in V_1$ and $a_1(j) = Q(a_2(j))$ if $a_2(j) \in V_2 \setminus V_1$. It is easy to verify that (V_1, a_1) is locally-optimal and the pairs (V_1, a_1) and (V_2, a_2) are nested. Consider the assignment a, which is defined the same way as a_1 by replacing P for Q. By the definition of Q, it follows that $\mathsf{cost}(V_1, a_1) \le \mathsf{cost}(V_1, a)$. Showing $c_{j,a(j)} \le 2c_{j,a_2(j)} + c_{j,d_1(j)}$ for all clients j and summing over all clients in C gives the required result.

If $a_2(j) \in V_1$ then $c_{j,a(j)} = c_{j,a_2(j)} \le 2c_{j,a_2(j)} + c_{j,d_1(j)}$. If $a_2(j) \in V_2 \setminus V_1$ then

$$\begin{aligned} c_{j,a(j)} = c_{j,P(a_2(j))} &\le c_{j,a_2(j)} + c_{a_2(j),P(a_2(j))} \\ &\le c_{j,a_2(j)} + c_{a_2(j),d_1(j)} \\ &\le 2c_{j,a_2(j)} + c_{j,d_1(j)}. \end{aligned}$$

The second inequality follows from the definition of $P(\cdot)$; the third is due to triangle inequality. □

LEMMA 3.4. *There exists a $(3, 2)$-augmentation for the hierarchical median problem. A $(3, 2\alpha)$-augmentation can be efficiently implemented, where $\alpha = 3 + \epsilon$.*

Proof. Given $(V_2, a_2) \in U$, with $\mathsf{ben}(V_2) < p$, let V with $\mathsf{ben}(V) = p$ be a solution to the p-median problem (with closest facility assignment). Using Lemma 3.2 we can find another set $V_1 \subset V_2$ with $|V_1| = |V|$ such that $\mathsf{cost}(V_1) \le \mathsf{cost}(V_2) + 2\mathsf{cost}(V)$. Using V_1 and (V_2, a_2) in Lemma 3.3 we get $\mathsf{cost}(V_1, a_1) \le \mathsf{cost}(V_1) + 2\mathsf{cost}(V_2, a_2)$. Since $\mathsf{cost}(V_2) \le \mathsf{cost}(V_2, a_2)$,

$$\begin{aligned} \mathsf{cost}(V_1, a_1) &\le \mathsf{cost}(V_1) + 2\mathsf{cost}(V_2, a_2) \\ &\le 2\mathsf{cost}(V) + \mathsf{cost}(V_2) + 2\mathsf{cost}(V_2, a_2) \\ &\le 3\mathsf{cost}(V_2, a_2) + 2\mathsf{cost}(V). \end{aligned}$$

Using an optimum solution (resp., an α-approximate solution [3]) to the k-median problem for V proves the first (resp., second) assertion. □

THEOREM 3.3. *There exists a solution to the hierarchical median problem with competitive ratio 24. A deterministic solution with competitive ratio 24α and a randomized solution with competitive ratio 10.76α can be computed efficiently, where $\alpha = 3 + \epsilon$.*

Proof. The existence and computability of chains of $\langle U, \preceq\rangle$ with the desired competitive ratios follow from Lemma 3.4 and Theorem 2.1. To convert a given chain $\mathcal{C}$ into a hierarchical sequence of solution pairs, we return the reverse of a maximal chain that contains $\mathcal{C}$. One can obtain a maximal chain of a given chain as follows: between any consecutive pair of solution pairs (S_0, a_0) and (S, a) such that $|S_0| = |S| + k$, for some $k > 1$, insert solution pairs (S_1, a_1), (S_2, a_2), $\ldots$, (S_{k-1}, a_{k-1}), where S_i equals $S_{i-1} \setminus \{f\}$ for an arbitrary f in $S_{i-1} \setminus S$ and a_i is identical to a_{i-1} except that all clients assigned to f in a_{i-1} are assigned to a facility in S_i that offers the least service cost. By the definition of competitive ratio (see Section 2.1), the competitive ratio of the chain is at least that of the hierarchical sequence, thus completing the proof of the theorem. □

REMARK 3.1. *In the proof of Lemma 3.4, we have made no additional assumptions about V_2 with respect to V or V_1. In fact, their costs are geometrically related. Using this fact, one can then do a better analysis of the incremental approximation algorithm and replace the performance ratios* $24, 24\alpha, 10.76\alpha$ *in Theorem 3.3, by* $20.71, 20.71\alpha$ *and* 10.03α *respectively. Details are deferred to the full paper.*

3.2.3 The incremental and hierarchical k-means problems

These problems are identical to their k-median counterparts, except that the k-means cost function is the the demand-weighted sum of squares of the distances of the clients to their nearest open facility. The set of solutions, their benefit, the binary operator, and the structure of posets are exactly the same as that of the k-median problems.

LEMMA 3.5. *There exists a* $(2, 8)$*-augmentation for the incremental k-means problem. Given an α-approximation algorithm to the k-means problem, a* $(2, 8\alpha)$*-augmentation can be computed.* □

THEOREM 3.4. *There exists a solution to the incremental k-means problem with competitive ratio* 64*. Given a polynomial-time α-approximation for the k-means problem, a deterministic solution and a randomized solution with competitive ratios* 64α *and* 31.82α*, respectively, can also be computed efficiently.* □

LEMMA 3.6. *There exists an* $(18, 8)$*-augmentation for the hierarchical k-means problem. Given an α-approximation algorithm for the k-means problem, an* $(18, 8\alpha)$*-augmentation can be computed.* □

THEOREM 3.5. *There exists a solution to the hierarchical k-means problem with competitive ratio* 576*. Given an α-approximation algorithm for the k-means problem, a deterministic and a randomized solution with competitive ratios* 576α *and* 151.1α *can be computed.* □

3.3 The incremental facility location problem

This problem was first defined by Plaxton [31], who also gives a $(4 + \epsilon)\alpha$ competitive algorithm, where α is the best available approximation factor for the facility location problem. We show that our framework also handles this problem with competitive ratio 4α.

The setting is similar to that of the k-median problem. In addition to the facility-set F, client-set C, client demands $w(\cdot)$, and metric connection cost $c(i, j)$ between any two locations i and j, there is an opening cost $v(i)$ for each facility i. To define the incremental facility location problem, we introduce a positive scaling factor λ, so that the total cost associated with opening a subset $Y \subseteq F$ is

$$\mathsf{cost}_\lambda(C, Y) = \lambda \cdot \sum_{j \in C} c(j, Y) \cdot w(j) + \sum_{i \in Y} v(i),$$

where $c(j, Y) = \min_{i \in Y} c(j, i)$. The incremental problem is to compute an ordered sequence of the facilities F, $(f_1, f_2, \cdots, f_n)$ and a *threshold sequence*[2] $t_1 \le t_2 \le \cdots \le t_n$ drawn from $\mathcal{R} \cup \infty$, such that for any scaling factor $\lambda > 0$, $\mathsf{cost}_\lambda(C, \{f_i | i \le k\})$ is a good approximation to $\mathsf{Opt}_\lambda = \min_{Y \subseteq F} \mathsf{cost}_\lambda(C, Y)$, where k is the smallest index such that $t_k \ge \lambda$.

To fit this problem into the framework described in Section 2, we can conceptually reformulate the problem as follows. Each solution element of the universe U is a subset of $F \times \mathcal{R}$. For $S \in U$, define $\mathsf{ben}(S) = \sup \pi_2(S)$, where π_i is the projection to the i^{th} coordinate, $i = 1, 2$. The binary operator is defined as $S_1 \preceq S_2$ iff $S_1 \subseteq S_2$. The cost function can now be defined as

$$\mathsf{cost}(S) = \mathsf{ben}(S) \cdot \sum_{j \in C} c(j, \pi_1(S)) \cdot w(j) + \sum_{i \in \pi_1(S)} v(i).$$

LEMMA 3.7. *There exists a* $(1, 1)$*-augmentation for the λ-facility location problem. A* $(1, \alpha)$*-augmentation can be computed efficiently, where* $\alpha = 1.52$*.* □

THEOREM 3.6. *There exists an incremental solution for the incremental facility location problem with competitive ratio* 4*. Moreover, an incremental solution of ratio* 4α *and a randomized solution of expected ratio* $e\alpha$ *can be computed efficiently, where* $\alpha = 1.52$*.*

Proof. Follows from Lemma 3.7 and Theorem 2.1. □

3.4 Incremental covering problems

3.4.1 The incremental k-set cover problem

Given a universe X of n elements and a collection of subsets of X, $\mathcal{C} = \{C_1, \cdots, C_m\}$ and a cost function $c : \mathcal{C} \to Q^+$, find an ordered sequence of $\mathcal{C}$, such that for *any* $k \in [1, n]$, the minimal prefix of the sequence that covers k elements is a good approximation to the k-set cover problem. Recall that the k-set cover problems asks for a min-cost subcollection of $\mathcal{C}$ that covers at least k elements.

In the language of Section 2, the universe U is $2^{\mathcal{C}}$. The benefit of $S \subseteq \mathcal{C}$ is simply the total number of elements covered by S. Then $S_1 \preceq S_2$ iff $S_1 \subseteq S_2$, and $\mathsf{cost}(S)$ is the sum of the weights of the subsets in S.

LEMMA 3.8. *There exists a* $(1, 1)$*-augmentation for k-set cover problem. Moreover, a* $(1, \alpha)$*-augmentation can be computed efficiently, where* $\alpha = \ln n + 1$*.* □

[2]The definition of threshold sequence in [31] is slightly different from ours, but serves the same purpose.

THEOREM 3.7. *There exists an incremental solution for the incremental k-set cover problem with competitive ratio 4. Moreover, a solution with ratio 4α can be computed efficiently, where $\alpha = \ln n + 1$.*

Proof. Follows from Lemma 3.8 and Theorem 2.1. □

3.4.2 The incremental k-vertex cover problem

Just as vertex cover is a special case of set cover, k-vertex cover problem is a special case of k-set cover. We hence have a corresponding incremental vertex cover problem. A 2-approximation algorithm for k-vertex cover is known [6, 15, 29].

THEOREM 3.8. *There exists an incremental solution for the incremental k-vertex cover problem with competitive ratio 4. Moreover, a solution with ratio 4α can be computed efficiently, where $\alpha = 2$.*

4 A general approach for problems with bounded envelope

Consider a problem Π specified by a quadruple $\langle U, \preceq, \mathsf{ben}, \mathsf{cost}\rangle$ as discussed in Section 2. We additionally assume that the range of benefit function is positive integers with maximum value B.

DEFINITION 4.1. *An α-approximate bounded envelope of a problem Π consists of values b_k for k ranging from 1 to B and solutions S_{n_i} with $\mathsf{ben}(S_{n_i}) = n_i$ for some $1 = n_1 < n_2 < \cdots < n_l = B$, such that:*

1. $b_k \le \mathsf{cost}(\mathsf{Opt}(k))$ *for* $1 \le k \le B$;
2. $\mathsf{cost}(S_{n_i}) \le \alpha \cdot b_{n_i}$ *for* $1 \le i \le l$;
3. $b_k = b_{n_{i-1}} + \frac{k-n_{i-1}}{n_i-n_{i-1}}(b_{n_i} - b_{n_{i-1}})$ *for* $n_{i-1} \le k \le n_i$ *and* $i = 2, \ldots, l$.

The idea of an α-approximate bounded envelope was first used for the k-MST and the minimum latency problems by Archer, Levin, and Williamson [2, 1].

DEFINITION 4.2. *An interpolation algorithm $\mathcal{I}$ for a problem Π is defined as an algorithm which when given two solutions S_{k_1} and S_{k_2} with $S_{k_1} \preceq S_{k_2}$, $\mathsf{ben}(S_{k_1}) = k_1$, $\mathsf{ben}(S_{k_2}) = k_2$, outputs a sequence $S_{k_1} \preceq S_{k_1+1} \preceq S_{k_1+2} \preceq \ldots \preceq S_{k_2-1} \preceq S_{k_2}$ of solutions such that for $k_1 < k < k_2$, we have $\mathsf{ben}(S_k) = k$, and $\mathsf{cost}(S_k) \le \mathsf{cost}(S_{k_1}) + \frac{k-k_1}{k_2-k_1}(\mathsf{cost}(S_{k_2}) - \mathsf{cost}(S_{k_1}))$.*

We use an idea similar to the generalized approach for incremental algorithms in Section 2 to get a good incremental algorithm for the problem Π given an α-approximate bounded envelope, interpolation algorithm and an augmentation algorithm $\mathsf{Augment}(S, p, \gamma, \delta)$ for that problem. The performance guarantee of the approximation algorithm in the augmentation is now replaced by α, the factor from the bounded envelope.

In particular, we obtain a 2-bounded envelope for the k-median problem, and an associated interpolation algorithm for both the incremental and hierarchical k-median problems, which allows us to replace the factors of $(3+\epsilon)$ in the competitive ratios of these problems coming from the approximation algorithm of Arya et al. [3] with a factor of 2. To do this, we exploit the fact that the facility location problem is a Lagrangean relaxation of the k-median problem, as observed by Jain and Vazirani [26]. We use a Lagrangean multiplier preserving (LMP) $(2-\epsilon)$-approximation facility location algorithm of Jain et al [25] to obtain a 2-approximate bounded envelope for the k-median problem. Our approximation algorithm computes the orderings for the incremental and hierarchical k-median problems using this 2-approximate bounded envelope, an interpolation algorithm, and the augmentation algorithms of Lemmas 3.2 and 3.4, respectively. Owing to space constraints, we only state the main results here and defer the algorithm details and analysis to the full version.

The above approach based on bounded envelopes yields a 16-competitive deterministic solution and a $4e$-competitive randomized solution for the incremental k-median problem, and a 41.42-competitive deterministic solution and a 20.06-competitive randomized solution for the hierarchical k-median problem.

5 Concluding Remarks

Our approach described in Section 2, and illustrated in Section 3, is general and can be easily used to handle other problems such as the k-center problem and the minimum dominating set problem. In Section 3.3, we have considered the incremental facility location problem introduced by [31]. Another natural incremental version of facility location can be defined using a partial facility location problem studied in [8], where all but s cities need to be served. Our approach again obtains an $O(1)$-competitive solution using a $O(1)$-approximation algorithm for the offline version.

One limitation of our work is that it may not lead to the best incremental solutions for a given problem. For instance, we can obtain an efficient 2-competitive algorithm for the unweighted vertex cover problem using the standard primal-dual approach (e.g., [34, Chap. 24]), while our generic approach only achieves a bound of 8. We also mention that for each of the problems discussed in the technical sections, there exists a constant c such that no c-competitive solution exists. For each of these problems, however, the best competitive ratio achievable is not known.

References

[1] A. Archer, A. Levin, and D. P. Williamson. A faster, better approximation algorithm for the minimum latency problem. Technical Report 1362, Cornell ORIE, 2003. Available at http://www.orie.cornell.edu/~dpw/minlat.ps.

[2] A. Archer and D. P. Williamson. Faster approximation algorithms for the minimum latency problem. In *ACM-SIAM SODA*, pages 88–96, 2003.

[3] V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala, and V. Pandit. Local search heuristics for k-median and facility location problems. *SIAM Journal on Computing*, 33:544–562, 2004.

[4] A. Blum, P. Chalasani, D. Coppersmith, B. Pulleyblank, P. Raghavan, and M. Sudan. The minimum latency problem. In *ACM STOC*, pages 163–171, 1994.

[5] A. Borodin and R. El-Yaniv. *Online Computation and Competitive Analysis.* Cambridge University Press, Cambridge, UK, 1998.

[6] N. H. Bshouty and L. Burroughs. Massaging a linear programming solution to give a 2-approximation for a generalization of the vertex cover problem. In *STACS*, volume 1373 of *LNCS*, pages 298–308, 1998.

[7] M. Charikar, C. Chekuri, T. Feder, and R. Motwani. Incremental clustering and dynamic information retrieval. *SIAM Journal on Computing*, pages 1417–1440, 2004.

[8] M. Charikar, S. Khuller, D. M. Mount, and G. Narasimhan. Algorithms for facility location problems with outliers. In *ACM-SIAM SODA*, pages 642–651, New York, Jan. 7–9 2001. ACM Press.

[9] M. Chrobak, C. Kenyon, J. Noga, and N. E. Young. Online medians via online bribery. http://www.arxiv.org/abs/cs.DS/0504103, Apr. 2005.

[10] M. Chrobak, C. Kenyon, and N. Young. The reverse greedy algorithm for the metric k-median problem. In *COCOON*, pages 654–660, 2005. Journal version to appear in IPL.

[11] B. Codenotti, G. De Marco, M. Leoncini, M. Montangero, and M. Santini. Approximation algorithms for a hierarchically structured bin packing problem. *Information Processing Letters*, 89:215–221, 2004.

[12] S. Dasgupta and P. Long. Performance guarantees for hierarchical clustering. *Journal of Computer and System Sciences*, 70:555–569, 2005.

[13] B. Dean, M. Goemans, and J. Vondrak. Approximating the stochastic knapsack problem: the benefit of adaptivity. In *IEEE FOCS*, 2004.

[14] A. Fiat and G. J. Woeginger, editors. *Online Algorithms: The State of the Art.* Springer, 1998.

[15] R. Gandhi, S. Khuller, and A. Srinivasan. Approximation algorithms for partial covering problems. In *ICALP*, pages 225–236, 2001.

[16] N. Garg. Saving an epsilon: A 2-approximation for the k-MST problem in graphs. In *ACM STOC*, pages 396–402, 2005.

[17] M. X. Goemans and J. Kleinberg. An improved approximation ratio for the minimum latency problem. *Mathematical Programming*, 82:111–124, 1998.

[18] T. González. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 2005.

[19] A. Gupta, M. Pal, R. Ravi, and A. Sinha. Boosted sampling: Approximation algorithms for stochastic optimization. In *ACM STOC*, pages 417–426, June 2004.

[20] A. Gupta, R. Ravi, and A. Sinha. An edge in time saves nine: LP rounding approximation algorithms for stochastic network design. In *IEEE FOCS*, 2004.

[21] J. Hartline and A. Sharp. Hierarchical flow. In *INOC*, pages 681–687, 2005.

[22] D. Hochbaum. The t-vertex cover problem: Extending the half integrality framework with budget constraints. In *APPROX*, pages 111–122, 1998.

[23] D. S. Hochbaum, editor. *Approximation Algorithms for NP-hard Problems.* PWS Publishing Company, Boston, MA, 1995.

[24] N. Immorlica, D. Karger, M. Minkoff, and V. Mirrokni. On the costs and benefits of procrastination: Approximation algorithms for stochastic combinatorial optimization problems. In *ACM-SIAM SODA*, January 2004.

[25] K. Jain, M. Mahdian, E. Markakis, A. Saberi, and V. V. Vazirani. Greedy facility location algorithms analyzed using dual-fitting with factor-revealing LP. *Journal of the ACM*, 50:795–824, 2003.

[26] K. Jain and V. V. Vazirani. Approximation algorithms for metric facility location and k-median problem using the primal-dual schema and Lagrangean relaxation. *Journal of the ACM*, 48:274–296, 2001.

[27] L. Jia, G. Lin, G. Noubir, R. Rajaraman, and R. Sundaram. Universal approximations for TSP, Steiner Tree, and Set Cover. In *ACM STOC*, pages 386–395, May 2005.

[28] M. Mahdian, Y. Ye, and J. Zhang. Improved approximation algorithm for metric facility location problems. In *APPROX*, pages 229–242, 2002.

[29] J. Mestre. A primal-dual approximation algorithm for partial vertex cover: making educated guesses. In *APPROX*, 2005.

[30] R. R. Mettu and C. G. Plaxton. The online median problem. *SIAM Journal on Computing*, 32:816–832, 2003.

[31] C. G. Plaxton. Approximation algorithms for hierarchical location problems. In *ACM STOC*, pages 40–49, 2003.

[32] P. Slavík. Improved performance of the greedy algorithm for partial cover. *Information Processing Letters*, 64:251–254, 1997.

[33] D. D. Sleator and R. E. Tarjan. Amortized efficiency of list update and paging rules. *Communications of the ACM*, 28(2):202–208, 1985.

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