An experimental evaluation of incremental and hierarchical k-median algorithms*

Chandrashekhar Nagarajan[1] and David P. Williamson[2]

[1] Yahoo! Inc., Sunnyvale, CA, 94089, USA, cn54@yahoo-inc.com
[2] Cornell University, Ithaca, NY, 14853, USA, dpw@cs.cornell.edu

Abstract. In this paper, we consider different incremental and hierarchical k-median algorithms with provable performance guarantees and compare their running times and quality of output solutions on different benchmark k-median datasets. We determine that the quality of solutions output by these algorithms for all the datasets is much better than their performance guarantees suggest. Since some of the incremental k-median algorithms require approximate solutions for the k-median problem, we also compare some of the existing k-median algorithms' running times and quality of solutions obtained on these datasets.

1 Introduction

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 [10].

The standard k-median problem has been the object of intense study in the algorithms community in recent years. Given the locations of a set of facilities and a set of clients in a metric space, and a parameter k, the *k-median problem* asks to find a set of k facilities to *open* such that the sum of the distances of the clients to the nearest open facility is minimized. Since the metric k-median problem is NP-hard [8], many researchers have focused on obtaining approximation algorithms for it. An α-approximation algorithm for a minimization problem runs in polynomial time and outputs a solution whose cost is at most α times the cost of the optimal solution. The factor α is sometimes called the *approximation factor* or *performance guarantee* of the algorithm. A solution for which the cost is at most α times the optimal cost is sometimes called *α-approximate*. The best approximation algorithm known for this problem has a performance guarantee of $3 + \epsilon$ and is due to Arya, Garg, Khandekar, Meyerson, Munagala and Pandit [2]; it is based on a local search heuristic.

* Supported in part by NSF grant CCF 0514528.

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. Mettu and Plaxton [10] gave a 29.86-competitive algorithm for the incremental k-median problem. Later Lin, Nagarajan, Rajaraman and Williamson [9] gave deterministic 16-competitive and randomized 10.88-competitive algorithms for the incremental k-median problem[3]. Their algorithms use either a k-median approximation algorithm or a Lagrangean Multiplier Preserving (LMP) facility location algorithm as a black box.

We also consider algorithms for the hierarchical k-median problem. In hierarchical clustering, we give clusterings with k clusters for all values of k by starting with each point in its own cluster and repeatedly merging selected pairs of clusters until all points are in a single cluster. We also consider a variation of this problem in which each cluster has a point designated as its center, and when we merge two clusters together to form a single cluster, one of the two centers becomes the center of the new 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. For the hierarchical k-median problem, the objective function for the k-clustering is its k-median cost; that is, the sum of the distances of each point to its cluster center. Plaxton [11] gave a 238.88-competitive algorithm for the problem. Lin et al. [9] later gave deterministic 40.42-competitive and randomized 20.06-competitive algorithms for the problem. Their algorithms again use either a k-median approximation algorithm or a LMP facility location algorithm as a black box.

In this paper, we consider the performance of these incremental and hierarchical k-median algorithms on different k-median benchmark datasets and compare their running times and quality of output solutions. Since the algorithms of Lin et al. require a k-median approximation algorithm or a LMP facility location algorithm as a black box, we also compare the performance of some of the existing k-median and LMP facility location algorithms. In particular, we implement five different k-median and LMP facility location algorithms. The first one is the single swap local search algorithm by Arya et al. [2], which gives 5-approximate solutions. We also consider the linear program (LP) rounding algorithm of Charikar, Guha, Tardos and Shmoys [4] which rounds the LP optimum to get 8-approximate solutions. Jain, Mahdian, Markakis, Saberi and Vazirani [7] give a greedy dual-fitting Lagrangean Multiplier Preserving (LMP) Facility Location (FL) algorithm which gives 2-approximate k-median solutions for some values of k. We also consider the standard k-median linear program

[3] Some of the results of Lin et al. were obtained independently by Chrobak, Kenyon, Noga, and Young [5].

and solve it optimally using CPLEX. The optimal solution can be fractional but still gives a good lower bound for the k-median problem. We also solve the k-median integer program optimally using CPLEX even though the algorithm is not polynomial time. These linear and integer programs give us bounds on the quality of the solutions of the other algorithms.

Given these algorithms, we implement several variants of the Lin et al. algorithms for the incremental k-median problem. We implement their algorithm using the Arya et al. local search algorithm for k-median, the Charikar et al. LP rounding algorithm for k-median, and the Jain et al. greedy algorithm which is an LMP algorithm for facility location. Additionally, we implement the original algorithm of Mettu and Plaxton for the incremental k-median problem. We are able to use the linear and integer programming solutions to bound the quality of the results we obtain.

We also implement several variants of the Lin et al. algorithms for hierarchical k-median problem. Again, we implement their algorithm using the Arya et al. local search algorithm for k-median, the Charikar et al. LP rounding algorithm, and the Jain et al. greedy algorithm. Additionally, we implement Plaxton's algorithm for the hierarchical k-median problem. Plaxton's algorithm requires an incremental k-median algorithm as a black box, and originally used the algorithm of Mettu and Plaxton as a subroutine. We implement this variant of Plaxton's algorithm, and also a variant that uses Lin et al.'s algorithm given the Arya et al. local search algorithm.

We test our algorithms on 43 different k-median instances drawn from the literature. In particular, we use forty instances from the OR Library [3], two instances from Galvão and ReVelle [6], and one instance from Alp, Erkut, and Drezner [1].

From the results we obtained we determine that all these algorithms perform much better in terms of quality of solution than their respective competitive/approximation ratios suggest. In particular, while we know of no polynomial-time algorithm with a competitive ratio better than 10 for the incremental and hierarchical median k-median problems, we typically obtained results which were within 10% of the k-median LP relaxation for incremental problems and 20% of the k-median LP relaxation for hierarchical k-median problems. We find this quite surprising in view of the strong constraints required on the structure of solutions for the incremental and hierarchical problems.

The algorithms of Mettu and Plaxton for incremental k-median and Plaxton for hierarchical k-median produce solutions that are not as good as those of Lin et al.; however, our implementation of the Mettu-Plaxton algorithm is significantly faster than our implementations of the Lin et al. algorithms, at least in part because the Lin et al. algorithms require approximate solutions of the k-median problem for all values of k.

Our paper is structured as follows. In Section 2, we sketch various algorithms we implemented. In Section 3, we discuss the datasets we used. In Section 4, we give the experimental results we obtained. In Section 5, we give our conclusions as well as some open problems prompted by our work. For space reasons, detailed statements of the algorithms and complete tables of results are omitted, and will appear in the full version of the paper.[4]

2 Algorithms

In this section, we discuss the various algorithms we implemented for the k-median, incremental k-median, and hierarchical k-median problems respectively.

2.1 The k-median problem

In this section we consider five different algorithms for the k-median problem: the single swap local search algorithm by Arya et al. [2]; the linear program (LP) rounding algorithm of Charikar et al. [4] which rounds the LP optimum to get an integer solution which is no more than 8 times the cost of the optimal LP solution; the Jain et al. [7] greedy dual-fitting Lagrangean Multiplier Preserving (LMP) Facility Location (FL) algorithm, which gives 2-approximate k-median solutions for some values of k; the standard k-median linear program, which we solve optimally using CPLEX; and the k-median integer program, which we also solve optimally using CPLEX even though the algorithm is not polynomial time. The optimal solution to the linear program can be fractional but still gives a good lower bound for the k-median problem. We now discuss each of these algorithms in turn. For space reasons, we cannot give full descriptions.

Local Search Algorithm of Arya et al. We consider the Arya et al.'s ([2]) single swap local search algorithm which computes a 5-approximate solution. The local search algorithm proceeds by starting with an arbitrary solution and repeatedly doing *valid swaps* on the current solution till no more valid swaps exist. A swap closes a facility in the current solution and opens a facility that was previously closed. A swap is considered valid if the cost of the new solution after swapping is less than the cost of the solution before swapping.

Arya et al. proved that the local search algorithm can be made to run in time polynomial in the input size by considering a swap as valid only if it improves the cost of the solution by a certain factor. However, for simplicity, we consider any cost-improving swap as a valid swap. We run this local search algorithm for each cardinality k. After this procedure we have locally optimal solutions for each value of k.

We do not implement the multi-swap (swaps involving more than one facilities) local search algorithm by Arya et al. because of its high running time even though it gives better approximation guarantee of $3+\epsilon$. We use the locally optimal solution of cardinality $k-1$ as a starting solution for the local search

[4] A more complete abstract of the paper, including full explanations of the algorithms, and full tables of results and running times, can be found at `http://www.orie.cornell.edu/~dpw/incexp.pdf`.

iteration for cardinality k. Since this solution is already a good solution for cardinality k we reduce the running times of the subsequent iterations. On average this improves the running times of local search by about 40%.

LP rounding algorithm of Charikar et al. We consider the LP rounding algorithm of Charikar et al. [4] which takes as input the fractional optimal solution of the standard LP relaxation $(k-P)$ of the k-median problem and produces an integer solution that is no more than 8 times the cost of LP optimum.

The algorithm is as follows. It starts with the optimal LP fractional solution for a particular value of k. First, the algorithm simplifies the problem instance by consolidating nearby clients and combining their demands such that the clients with nonzero demands are far from each other the resulting problem instance. It then simplifies the structure of the optimal fraction solution by consolidating nearby fractional facilities. The resulting solution has nonzero fractional value only on facilities with nonzero demands and the LP variables for the facilities are no less than $\frac{1}{2}$. The algorithm then modifies this solution to a solution where the LP variables for the facilities take values of only 0, $\frac{1}{2}$ and 1. It then opens no more than k of these facilities, selecting them based on their distance to other facilities with positive LP value.

Greedy LMP FL Algorithm of Jain et al. Jain et al. [7] give a LMP greedy dual-fitting algorithm for the facility location problem. In this algorithm, we maintain a dual value v_j for every client which is its total contribution to getting connected to a open facility. Some part of this dual v_j pays for the j's connection cost and the remainder is paid toward facility opening costs. We increase the duals of the clients uniformly and open a facility when a facility has enough contribution from the clients to match the facility opening cost. We say a client is connected to a facility if the connection cost is paid for by its dual value. We stop increasing the dual for a client if it is connected to a open facility.

Since this facility location algorithm is a LMP 2-approximation algorithm for the facility location (FL) problem, we can obtain something called a *bounded envelope* for the k-median problem as described in Lin et al. [9]. The bounded envelope gives 2-approximate solutions for the k-median problem for some values of k as well as a corresponding piecewise linear lower bound on the values of k-median solutions for all values of k, where the breakpoints of the lower bounds occur at values of k for which we have 2-approximate solutions. Lin et al. give a procedure for computing the bounded envelope given the LMP FL algorithm.

Solving Linear Program using CPLEX. We solve the linear programming relaxation $(k-P)$ of the standard k-median problem using the CPLEX solver.

To speed up the running time of the linear program solver, we tried to give the optimal solution of $(k-1)^{th}$ run as an initial starting solution to the iteration of cardinality k for all values of k. But there was no significant improvement of the running times of the linear programs on average.

Solving Integer Program using CPLEX. We solve the integer program $(k{-}IP)$ optimally using the CPLEX solver; $(k{-}IP)$ is the same as $(k{-}P)$ except that we require the decision variables to be 0-1. The CPLEX solver provides a way to give a good initial guess to the solver so that it can prune many low quality solutions. We give the optimal integer solution with $k-1$ facilities as an initial guess for the CPLEX integer program iteration with cardinality k. As the optimal solution for the k-median problem for a smaller value of cardinality is a feasible solution for the k-median problem with larger cardinality, the initial guess is feasible. Even though this makes the solver find the optimal integral solution faster in some cases, it does not work in all cases and on average the improvement in running time is not significant.

2.2 Incremental *k*-median

In this section we briefly explain the Mettu and Plaxton's incremental k-median algorithm and Lin et al.'s incremental k-median algorithm.

Mettu and Plaxton's Algorithm. Mettu and Plaxton's [10] incremental k-median algorithm uses a hierarchical greedy approach to choose the next facility in the incremental order to be opened. The basic idea behind this approach is as follows. Rather than selecting the next point in the ordering based on a single greedy criterion, they greedily choose a region and then recursively choose smaller regions till they arrive at a single facility which then becomes the next facility to open. Thus the choice of the next facility is influenced by a sequence of greedy criteria addressing successive finer levels of granularity.

Lin et al.'s incremental *k*-median algorithm. We implement the incremental algorithm AltIncApprox of Lin et al. [9] for the incremental k-median problem on these datasets. We use Arya et al.'s local search algorithm with single swaps and the LP rounding technique of Charikar et al. to generate good k-median solutions for all possible k for each of these datasets. We bucket these solutions into buckets of geometrically increasing cost. We take the costliest solution from each bucket. We then consider each of these solutions in order of decreasing number of medians, and use each such solution to find another solution with the same number of medians that is contained with the next larger solution. This gives us a sequence of k-median solutions such that any smaller solution is a subset of any larger solution. This sequence of solutions gives a natural ordering of the facilities.

We also implement the incremental algorithm BoundedIncApprox of Lin et al. [9] using the k-median bounded envelope obtained by running the Jain et al. algorithm on the datasets. By using the 2-approximate solutions obtained from this algorithm for some values of k, we can apply the procedure given above to obtain an ordering of the facilities.

2.3 Hierarchical k-median

We test the hierarchical k-median algorithms of Lin et al. [9] against the previously known hierarchical k-median algorithm by Plaxton [11].

Plaxton's Algorithm. Plaxton's algorithm takes in an incremental k-median solution as input and finds a *parent* function for each facility this incremental ordering. A hierarchical k-median solution obtained from an ordering can be considered as solutions obtained by repeatedly closing the last open facility in ordering and assigning its clients to an earlier facility. This mapping is exactly captured by the parent function in the Plaxton's algorithm. A parent function for an ordering maps every facility in the order to a facility that is earlier in the ordering. The parent of a facility is the facility that its clients will get assigned to when the facility is closed.

Plaxton's parent function is assigned as follows: Given an incremental k-median solution to the problem, a parent is assigned to every facility in the reverse order of the incremental solution. The parent of a facility f is determined by the earliest facility in the ordering that is either the closest facility or satisfies a certain equation. The equation essentially finds a facility whose distance to f is no more than the average distance of f's clients to f.

We run the Plaxton's parent function algorithm on the incremental k-median solutions given by running the Mettu and Plaxton's algorithm and AltIncApprox algorithm using Arya et al.'s local search solutions on the datasets.

Lin et al.'s hierarchical k-median algorithm. We run the generic algorithm AltIncApprox of Lin et al. [9] for the hierarchical k-median problem on the datasets using different k-median algorithms as black box. We use Arya et al.'s local search algorithm and Charikar et al.'s LP rounding algorithm to generate good k-median solutions. We also implement the incremental algorithm BoundedIncApprox of Lin et al. [9] using the k-median bounded envelope obtained by running Jain et al. algorithm on the datasets. As in the incremental k-median algorithm of Lin et al. we must find approximate solutions to the k-median problem, which we then put in buckets of geometrically increasing cost, then take the costliest solution from each bucket. We consider these solutions in order of decreasing size, and use each solution to find a k-clustering that is consistent with a hierarchical clustering on the larger solutions already considered.

3 Datasets

In our experiments we use these following datasets for the comparison of k-median, incremental k-median and the hierarchical k-median algorithms.

1. *OR Library:* These 40 datasets of the uncapacitated k-median problems are part of the OR Library [3], which is a collection of test datasets for a variety

of OR problems created by J. E. Beasley. These 40 test problems are named *pmed*1, *pmed*2, ..., *pmed*40 and their sizes range from $n = 100$ to 900. As noted in [3], we apply Floyd's algorithm on the adjacency cost matrix in order to obtain the complete cost matrix.

2. *Galvão:* This set of instances (*Galvão100* and *Galvão150*) is obtained from the work of Galvão and ReVelle [6]. Even though the sizes of these datasets are small ($n = 100$ and $n = 150$), the integrality gaps for some values of k (number of medians) are larger than traditional datasets.
3. *Alberta:* This dataset is generated from a 316-node network using all population centers in Alberta (see Alp, Erkut and Drezner [1]) where the distances are computed using the shortest path metric on the actual road network of Alberta.

4 Experimental Results

4.1 The *k*-median problem

In this section we compare the performance in terms of running times and quality of solutions of five different algorithms on the datasets described: CPLEX solver for the k-median linear program, CPLEX solver for k-median integer program, Arya et al.'s single swap local search algorithm, Charikar et al.'s LP rounding algorithm and the bounded envelope of Jain et al.'s greedy algorithm. All experiments were done on machines with Intel Core 2 2.40GHz processor with 2 gigabytes of physical memory. The linear programs and integer programs on the data sets are solved using CPLEX Version 10.1.0. The Arya et al.'s single swap local search algorithm and Jain et al. algorithm are solved using MATLAB version 7.0. The tolerance for the bounded envelope that we use for the termination of binary search is 0.01 (see Lin et al. [9] for the bounded envelope procedure). For space reasons, we cannot present the full table of results; however, Figures 1 and 2 show how the costs of the k-median solutions from the integer optimum, Arya et al.'s local search algorithm, Charikar et al.'s LP rounding algorithm and the Jain et al.'s greedy algorithm compare to the linear program for different values of k for two sample datasets *pmed*40 and *Galvão150*. This performance was typical.

Even though the Arya et al.'s algorithm's performance guarantee is 5, in practice the local search algorithm performs much better than that. The local optimums are within 1% from the linear program optimum on average. Charikar's et al.'s LP rounding algorithm performs even better as most of the LP solutions are already integral or very close to being integral except for some small values of k. Note that the Jain et al.'s greedy LMP FL algorithm gives only a bounded envelope and does not give k-median solutions for all values of k. Here we can see that the LP rounding algorithm and the local search algorithm perform better than Jain et al.'s algorithm.

In terms of running time, the LP solver runs faster than the local search and greedy algorithm for all datasets. Also the IP solver takes a lot more time to solve all the instances of k for bigger datasets.

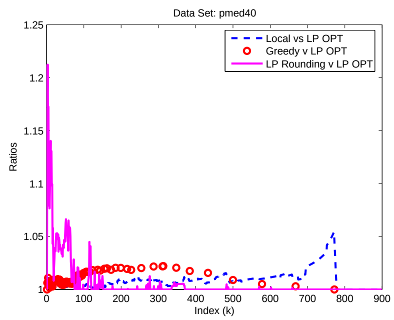

Fig. 1. Quality of solutions of k-median algorithms (dataset *pmed*40)

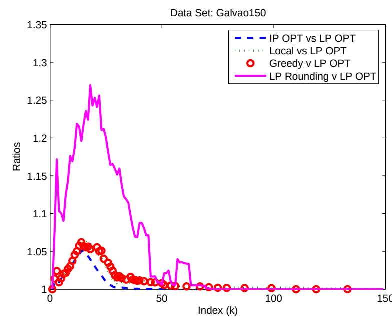

Fig. 2. Quality of solutions of k-median algorithms (dataset *Galvão150*)

4.2 Incremental k-median

In this section we compare the performances of four different incremental k-median algorithms on the selected datasets: Mettu and Plaxton's incremental k-median algorithm (MPInc), Lin et al.'s AltIncApprox algorithm with solutions from the Arya et al.'s single swap local search algorithm (LInc) and Charikar et al.'s LP rounding (LPR) and Lin et al.'s BoundedIncApprox algorithm with the bounded envelope obtained from the Jain et al.'s greedy LMP FL algorithm (GInc).

Our experiments show that Lin et al.'s algorithms perform much better than the Mettu and Plaxton's algorithm on the datasets. This inference is reinforced by Figures 3, 4, 5, and 6 which show that the ratios of the costs of solutions obtained from Lin et al.'s incremental algorithms to the LP optimum are always better than the corresponding ratios of Mettu and Plaxton's algorithm for a sample of datasets (*pmed*10, *pmed*25, *pmed*40 and *Galvão150*). The Mettu-Plaxton algorithm runs much faster than Lin et al.'s algorithms; these use a k-median algorithm or a bounded envelope algorithm as a blackbox, which make them very slow. However the quality of the incremental solutions obtained from Lin et al.'s algorithm is much better than that of the Mettu-Plaxton algorithm.

4.3 Hierarchical k-median

In this section we compare the performance of Plaxton's hierarchical k-median algorithm against Lin et al.'s AltIncApprox hierarchical k-median algorithm on the datasets. Note that Plaxton's algorithm takes in any incremental k-median solution as input and outputs a parent function which defines the hierarchical solution. We give the incremental k-median solutions from our runs of AltIncApprox and the Mettu-Plaxton algorithms as input to the Plaxton's hierarchical algorithm (PHLI and PHMP) and compare them against Lin et al.'s hierarchical k-median algorithms' solutions (HL, HG and LPRH) for different datasets.

Figures 7, 8, 9, and 10 show how the costs of the hierarchical k-median solutions for different algorithms compare against the optimal linear program

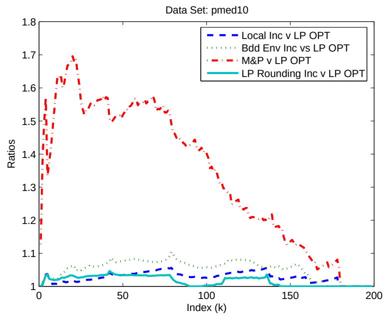

Fig. 3. Quality of solutions of incremental k-median algorithms (dataset *pmed*10)

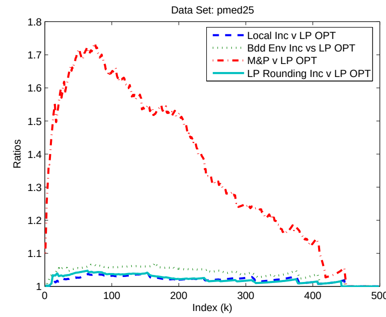

Fig. 4. Quality of solutions of incremental k-median algorithms (dataset *pmed*25)

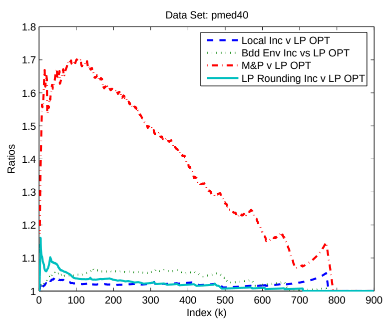

Fig. 5. Quality of solutions of incremental k-median algorithms (dataset *pmed*40)

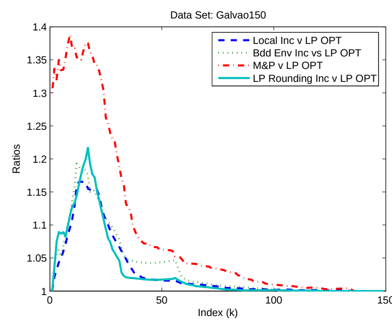

Fig. 6. Quality of solutions of incremental k-median algorithms (dataset *Galvão150*)

solutions for different values of k for sample datasets *pmed*10, *pmed*25, *pmed*40 and *Galvão150*. The algorithms we consider are AltIncApprox algorithm (using Arya et al.'s local search k-median solutions (HL) and Charikar et al.'s LP rounding solutions (LPRH)), BoundedIncApprox algorithm (using bounded envelope from Jain et al.'s greedy algorithm) (HG), Plaxton's hierarchical k-median algorithm on the incremental solutions of AltIncApprox algorithm (PHLI) and Plaxton's algorithm on Mettu and Plaxton's incremental k-median solutions (PHMP).

The hierarchical solutions obtained by AltIncApprox algorithms are better than other algorithms. The ratios for the PHMP algorithm are not as good as for the other algorithms since PHMP uses the incremental k-median solutions of Mettu and Plaxton as input which are not as good as other incremental algorithms in terms of quality. Lin et al.'s hierarchical algorithm (HL) which computes hierarchical solutions directly from k-median solutions performs better

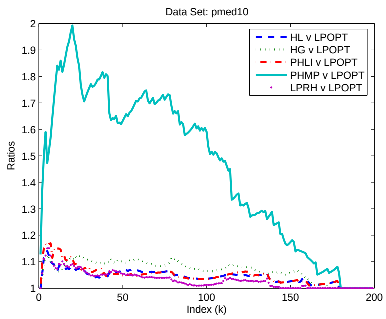

Fig. 7. Quality of solutions of hierarchical k-median algorithms (dataset $pmed10$)

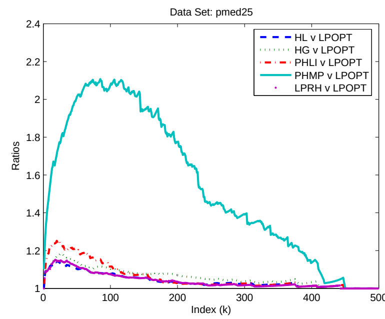

Fig. 8. Quality of solutions of hierarchical k-median algorithms (dataset $pmed25$)

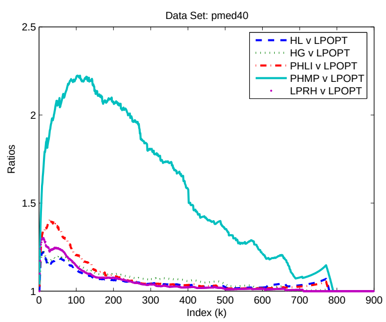

Fig. 9. Quality of solutions of hierarchical k-median algorithms (dataset $pmed40$)

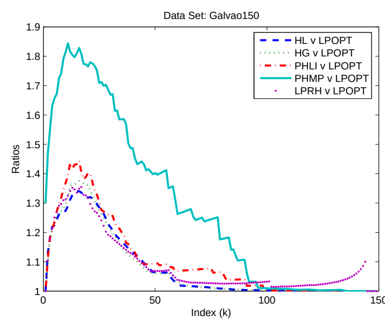

Fig. 10. Quality of solutions of hierarchical k-median algorithms (dataset *Galvão150*)

than the Plaxton's hierarchical algorithm even when the incremental solutions from AltIncApprox are given as input.

5 Conclusions

We evaluate different k-median, incremental k-median and hierarchical k-median algorithms on different datasets and show our results here. For the k-median problem, Charikar et al.'s LP rounding algorithm performs better and faster on average than other k-median algorithms like Arya et al.'s local search algorithm. We also notice that in many real-life datasets the optimal LP solution for the k-median problems for most values of k are integers which also makes the LP rounding techniques much better in terms of the quality of the solutions.

The quality of incremental solutions, when AltIncApprox algorithm is run on the k-median solutions of Arya et al.'s local search algorithm and Charikar et al.'s LP rounding algorithm, are much better than the incremental solutions of Mettu and Plaxton's algorithm. Even though the LP rounding algorithm performs poorly for some small values of k, Lin et al.'s incremental and hierarchical algorithms skips many of these poor solutions while bucketing the solutions geometrically and this makes the corresponding incremental solutions comparable in quality to the incremental solutions obtained from Arya et al.'s local search k-median solutions.

Mettu and Plaxton's incremental k-median algorithm is much faster than the other incremental k-median algorithms we implement. However one important point to note here is that we find good k-median solutions for all values of k both in Arya et al.'s local search algorithm and Charikar et al.'s LP rounding algorithm. Most of these solutions are not used at all by the Lin et al. algorithms since it uses only one solution from each of the geometrically increasing buckets. It would be useful if we would somehow be able to find a sequence of k-median solutions that are geometrically increasing in cost in a faster way; this could lead to significant improvements in the running times of the Lin et al. algorithms.

References

1. Alp, O., Erkut, E., Drezner, D.: An efficient genetic algorithm for the p-median problem. Annals of Operations Research 122, 21–42(22) (2003)
2. Arya, V., Garg, N., Khandekar, R., Meyerson, A., Munagala, K., Pandit, V.: Local search heuristics for k-median and facility location problems. SIAM Journal on Computing 33, 544–562 (2004)
3. Beasley, J.E.: A note on solving large p-median problems. European Journal of Operational Research 21, 270–273 (1985)
4. Charikar, M., Guha, S., Tardos, E., Shmoys, D.B.: A constant-factor approximation algorithm for the k-median problem (extended abstract). In: Proceedings of the 31th Annual ACM Symposium on Theory of Computing. pp. 1–10 (1999)
5. Chrobak, M., Kenyon, C., Noga, J., Young, N.E.: Incremental medians via online bidding. Algorithmica 50, 455–478 (2008)
6. Galvao, R.D., ReVelle, C.: A Lagrangean heuristic for the maximal covering location problem. European Journal of Operational Research 88(1), 114–123 (1996)
7. Jain, K., Mahdian, M., Markakis, E., Saberi, A., Vazirani, V.: Greedy facility location algorithms analyzed using dual fitting with factor-revealing LP. Journal of the ACM 50, 795–824 (2003)
8. Kariv, O., Hakimi, S.: An algorithm approach to network location problems ii. the p-medians. SIAM Journal of Applied Mathematics 37, 539–560 (1979)
9. Lin, G., Nagarajan, C., Rajaraman, R., Williamson, D.P.: A general approach for incremental approximation and hierarchical clustering. SIAM Journal on Computing 39, 3633–3669 (2010)
10. Mettu, R.R., Plaxton, C.G.: The online median problem. SIAM Journal on Computing 32, 816–832 (2003)
11. Plaxton, C.G.: Approximation algorithms for hierarchical location problems. Journal of Computer and System Sciences 72, 425–443 (2006)