Budgeted Red-Blue Median and Its Generalizations*

MohammadTaghi Hajiaghayi[1], Rohit Khandekar[2], and Guy Kortsarz[3,**]

[1] AT&T Labs–Research & University of Maryland
hajiagha@research.att.com
[2] IBM T.J. Watson research center
rohitk@us.ibm.com
[3] Rutgers University–Camden
guyk@camden.rutgers.edu

Abstract. In a Content Distribution Network application, we have a set of servers and a set of clients to be connected to the servers. Often there are a few server types and a hard budget constraint on the number of deployed servers of each type. The simplest goal here is to deploy a set of servers subject to these budget constraints in order to minimize the sum of client connection costs. These connection costs often satisfy metricity, since they are typically proportional to the distance between a client and a server within a single autonomous system. A special case of the problem where there is only one server type is the well-studied *k-median* problem.

In this paper, we consider the problem with two server types and call it the *budgeted red-blue median* problem. We show, somewhat surprisingly, that running a single-swap local search for each server type *simultaneously*, yields a constant factor approximation for this case. Its analysis is however quite non-trivial compared to that of the k-median problem (Arya et al., 2004; Gupta and Tangwongsan, 2008).

Later we show that the same algorithm yields a constant approximation for the *prize-collecting* version of the budgeted red-blue median problem where each client can potentially be served with an alternative cost via a different vendor. In the process, we also improve the approximation factor for the *prize-collecting k-median* problem from 4 (Charikar et al., 2001) to $3+\epsilon$, which matches the current best approximation factor for the k-median problem.

1 Introduction

Consider the following problem called the *budgeted red-blue median problem*. The input is a set of facilities $\mathcal{F}$ and a set of clients $\mathcal{C}$ in a metric space. The distance between two points in this metric space $i, j \in \mathcal{F} \cup \mathcal{C}$ is denoted by $d(i, j)$. The

* Part of this work was done while the authors were meeting at DIMACS. We would like to thank DIMACS for hospitality.
** Research partially supported by NSF grant 0819959.

M. de Berg and U. Meyer (Eds.): ESA 2010, Part I, LNCS 6346, pp. 314–325, 2010.

facilities are partitioned into two sets: *red* facilities $\mathcal{R}$ and *blue* facilities $\mathcal{B}$. The input also includes two integers $k_r, k_b > 0$. Given a subset of *open* facilities, a client j gets served by the nearest open facility. The goal of the problem is to open a subset of red facilities $R \subseteq \mathcal{R}$ and a subset of blue facilities $B \subseteq \mathcal{B}$ such that

- $|R| \leq k_r$ and $|B| \leq k_b$,
- the total connection cost $\mathbf{cost}(R, B) := \sum_{j \in \mathcal{C}} d(j, R \cup B)$ is minimized.

Here $d(j, S) = \min_{i \in S} d(j, i)$ denotes the shortest distance from j to any point in S. A special case in which all facilities have the same color is the well-studied *k-median problem*. In content distribution network applications and several other applications in telecommunication networks and clustering, it is vital to obtain solutions for these problems without violating any budget (see e.g., [5,3]).

Another related and well-motivated problem is the *weighted W-median* problem in which given a non-negative opening cost w_i for each facility i, we want to open a set of facilities whose opening cost is within our budget W and minimize the total connection cost. This budget constraint is a Knapsack constraint and thus for general opening costs, we do not hope to get any approximation algorithm if we insist not to violate our budget W. Indeed as in Knapsack, if we are allowed to violate the budget within a factor $1+\epsilon$, we can obtain a constant factor approximation algorithm using the filtering method of Lin and Vitter [21]. In addition, for polynomially bounded opening costs (for which Knapsack is solvable), we can solve the problem on trees without violating budget W. Using probabilistic embeddings of general metrics into tree metrics [4,11]), this immediately results in an $O(\log n)$ approximation algorithm for weighted W-median in general metrics without violating budget W when opening costs are polynomially bounded. When there are only two different facility opening costs, one can guess the number of facilities of each type in the optimum solution. Thus this special case can be reduced to the budgeted red-blue median problem.

Last but not least, in several of the above applications, each client can be satisfied with an alternative cost often via a different vendor. Indeed, this cost is called the *penalty* of this client that we pay in case it is not connected to one of our deployed servers. More formally, in all problem formulations above we can assume that each client $j \in \mathcal{C}$ has a *penalty* $p_j \in \mathbf{Q}_+$. The client pays the service cost, i.e., its distance to the nearest open facility, if it is at most its penalty p_j; otherwise the client remains unserved and pays the penalty p_j. The goal then is to minimize the sum of connection costs and paid penalties.

1.1 Related Results

Aforementioned the most special case of our problems in this paper is the well-known k-median problem. The first constant factor approximation for the k-median problem was given by Charikar et al. [7], which was subsequently improved by Jain-Vazirani [17], Charikar-Guha [6], and Arya et al. [3]. The latter presents the current best approximation factor of $3 + \epsilon$ for k-median via a local search heuristic. Their analysis was recently simplified by Gupta and

Tangwongsan [14]. The problem cannot be approximated within a factor strictly less than $1 + 2/e$, unless $\mathbf{NP} \subseteq \mathbf{DTIME}[n^{O(\log \log n)}]$ [16]. It is known that the integrality gap of the natural LP relaxation of the problem is at most 3, but currently there is no algorithm that achieves a 3-approximation in polynomial time [1]. An extension of k-median to the case in which we can open at most k facilities, but also have to pay their facility opening cost was studied by [10], who gave a 5-approximation. The k-median problem with penalties was also considered; the current best approximation factor for prize-collecting k-median is 4 due to Charikar et al. [8]. The problem in which the underlying metric is Euclidean, although NP-hard [23], admits a PTAS due to the results of Arora, Raghavan, and Rao [2], and then Kolliopoulos and Rao [18] (who provided an almost-linear time algorithm).

The Lagrangian relaxation approach was used by Jain and Vazirani [17] for the k-median problem. When we apply this approach to the budgeted red-blue median problem, we can get two solutions whose convex combination has cost at most a constant factor times the optimum cost. These two solutions have k_r^1 (resp. k_r^2) red and k_b^1 (resp. k_b^2) blue facilities where $k_r^1 + k_b^1 = k_r^2 + k_b^2 = k_r + k_b$. It may happen, for example, that $k_r^1 > k_r$ and $k_b^2 > k_b$, i.e., the bound on red facilities is violated in the first solution and the bound on blue facilities is violated in the second solution. Unlike the case for the k-median, both of these solutions may be infeasible. Therefore, it seems very hard to combine them to get a solution that has no violation while having cost within a constant factor of the optimum cost. The observation that the Lagrangian relaxation approach fails for the budgeted red-blue median problem was also shared and verified by Jain [15].

Local search based approaches. From a practical point of view, a simple combinatorial algorithm is much more desirable than the one that requires to solve a linear programming relaxation. To this end, our main approach in this paper is to extend the local search technique which is a popular heuristic for hard combinatorial optimization problems. A relatively few instances of approximation guarantees via local search are known. Korupolu, Plaxton, and Rajaraman [19] gave the first approximation guarantees of this type for the facility location and k-median problems based on a simple local search heuristic proposed by Kuehn and Hamburger [20]. For the k-median problem, however, they violate the constraint on the number of open facilities by a factor $1 + \epsilon$. Later Arya et al. [3] could approximate the problem without violating this constraint. The local search later has been used for other facility location type problems [22,25,9,24] and recently even for maximum generalized assignment [13] and maximizing submodular functions [12].

1.2 Our Results

The main result of this paper is a constant factor approximation algorithm for the budgeted red-blue median problem via novel analysis of a natural local search algorithm. More formally, we analyze the following local search algorithm.

1. Let $R \subset \mathcal{R}$ and $B \subset \mathcal{B}$ be *arbitrary* subsets with $|R| = k_r$ and $|B| = k_b$.
2. While there exist $r \in R$, $r' \in \mathcal{R}$ and $b \in B$, $b' \in \mathcal{B}$ such that $\mathbf{cost}(R - r + r', B - b + b') < \mathbf{cost}(R, B)$ do: $R \leftarrow R - r + r'$ and $B \leftarrow B - b + b'$.
3. Output R and B.

Here $S - s_1 + s_2$ denotes $(S \setminus \{s_1\}) \cup \{s_2\}$. Since r and r' (or b and b') may be identical, our algorithm outputs a locally optimum solution w.r.t. three local operations: (1) delete a red facility and add a red facility, (2) delete a blue facility and add blue facility, and (3) delete a red and a blue facilities and add a red and a blue facilities. In Section 2, we prove the following theorem.

Theorem 1. *The above local search algorithm yields a constant approximation to the budgeted red-blue median problem.*

In fact, it is somewhat surprising that this natural local search algorithm even works. We point out why in the next section by explaining the main challenges in the analysis. We omit the standard details regarding how to make this algorithm run in polynomial time. In Section 3, we show how the local search analysis can be extended to the prize-collecting version. More specifically, we improve the current best approximation factor 4 of the LP-rounding algorithm for prize-collecting k-median due to Charikar et al. [8] as follows.

Theorem 2. *The multi-swap local search algorithm of Arya et al. [3] yields $(3 + \epsilon)$-approximation for the prize-collecting k-median problem.*

Last but not least, we show how we can combine the techniques for the budgeted red-blue median problem and prize-collecting variants to obtain the most general theorem of this paper.

Theorem 3. *There is a local search algorithm that yields a constant approximation for the prize-collecting budgeted red-blue median problem.*

The proof of this theorem is omitted from this extended abstract due to lack of space.

1.3 An Overview of Our Techniques

The budgeted red-blue median problem. Let us first understand why the standard local search analysis of k-median [3,14] does not extend easily to the budgeted red-blue median problem. In the k-median analysis, we consider several test swaps for the locally optimum solution S. Each of these swaps includes deleting a facility from S and adding a facility from the optimum solution O and rerouting the clients. These swaps are chosen carefully to bound the cost of S. In case of the budgeted red-blue median problem, however, this choice may conflict with the budget constraints on the number of red and blue facilities allowed. For example, after deleting, say, a red facility, to keep the cost bounded, one may need to add a blue facility to serve the clients previously served by the dropped red facility.

This happens, for example, when there is no other red facility close-by. In such a case, we are forced to delete another blue facility and possibly add another red facility in order to balance the number of red and blue facilities. As a result, bounding the cost of the solution after the swap becomes much trickier.

Our analysis begins by partitioning the solutions S and O into *blocks* (see Section 2.1) with some useful structural properties. Intuitively speaking, a block is a subset of $S \cup O$ for which the test swaps can be analyzed "independently" of other blocks, even when a test swap involves rerouting clients served by facilities from multiple blocks. These blocks are defined based on the distances and the colors of the facilities. For example, let $s_i \in S$ be the closest facility in S for exactly one facility $o_i \in O$ for $i = 1, 2$. If s_i has the same color as o_i, then $\{s_i, o_i\}$ defines a block. On the other hand if $\{s_1, s_2, o_1, o_2\}$ has two red and two blue facilities, this set defines a block. In general, a block contains an equal number of red facilities and an equal number of blue facilities from the two solutions S and O such that for any facility $o \in O$ in a block, the closest facility to it in S is also in the same block. A typical block also satisfies a key property: it contains a large number of facilities in S that are not the closest facilities in S to any facility in O. It turns out that such facilities, called *very good facilities*, are compatible to be swapped with any facility in O [14] and their abundance is crucial to the overall analysis. We use a careful counting argument to show that a partitioning into blocks satisfying these properties exists.

In Section 2.2, we describe the test swaps for any single block. If s_i and o_i described above have the same color, we can consider the swap: add o_i and delete s_i. However, if $s = s_i$ is the closest one to several facilities $\{o_1, \ldots, o_l\}$ in the optimum solution, then deleting s may be bad for our solution. The previous k-median analyses, therefore, avoided swaps in which s is deleted.

Unfortunately, it turns out that we do not have a luxury of avoiding such swaps. Consider, for example, the case where $k_r = 1$ and s is the only red facility in S. Suppose that o is the unique red facility in O. To bound the cost of clients served by o in solution O, we need to consider a test swap in which o is added. Note however that if o is added, s must be deleted to satisfy the budget $k_r = 1$. Our analysis considers a test swap in which we delete s and open the facility $o_i \in \{o_1, \ldots, o_l\}$ that is closest to s. If s and o_i are of different colors, we combine this swap with another carefully chosen red-blue swap to balance the number of red and blue facilities. The cost after such a swap may potentially be significantly higher than that of the optimum solution. To "cancel" this high cost, we consider several other test swaps in which facilities $\{o_1, \ldots, o_l\}$ are added one-by-one. Using the properties of a block mentioned above, we show how to bound the overall cost for all the swaps considered.

In our opinion, these new swaps and a method to bound their costs is the main technical contribution of our paper. We encourage the reader to read the exposition in paragraphs titled 'Intuition' and 'Example in Figure 2' in Section 2.3 for further intuition behind our approach.

The prize-collecting version. We show that the multi-swap local search algorithm of the k-median problem [3] yields $(3+\epsilon)$-approximation for the prize-collecting

k-median problem. The proof is based on the techniques of Arya et al. [3] or Gupta and Tangwongsan [14] applied to the clients that do not pay penalty in either solution S or O. The other clients contribute the same amount to either solutions and thus are easy to handle. Essentially the same line of argument holds for the prize-collecting version of budgeted red-blue problem.

2 Proof of Theorem 1

We begin with some notation and preliminaries. We call the local search operations in our algorithm as *valid swaps*. Let $O = R^* \cup B^*$ denote the optimum solution where $R^* \subset \mathcal{R}$ and $B^* \subset \mathcal{B}$ and let $S = R \cup B$ denote the locally optimum (also called local) solution. For a facility $o \in O$, let $N^*(o)$ denote the clients that are served by o in solution O, i.e., these clients have o as the closest facility among facilities in O. Similarly, for $s \in S$, let $N(s)$ denote the clients that are served by s in solution S. For $A \subset O$, let $N^*(A) = \cup_{o \in A} N^*(o)$ and for $A \subset S$, let $N(A) = \cup_{s \in A} N(s)$. For a client $j \in \mathcal{C}$, let $O_j = d(j, O)$ and $S_j = d(j, S)$ be its contribution to the optimum and local solutions respectively.

Definition 1 (functions η and μ). *Define a function $\eta : O \to S$ as follows. For $o \in O$, let $\eta(o)$ be the facility in S that it closest to o, where ties are broken arbitrarily. Thus we have $d(o, \eta(o)) = d(o, S)$.*

For a facility $s \in S$ with $\eta^{-1}(s) \neq \emptyset$, define $\mu(s)$ to be the facility in $\eta^{-1}(s)$ that it closest to s where ties are broken arbitrarily, i.e., we have $d(s, \mu(s)) = d(s, \eta^{-1}(s))$.

See Figure 2 for an example. Note that if $o \in O \cap S$, then we have $\eta(o) = o$. The definition of function η is motivated by the paper of Gupta and Tangwongsan [14] who offer a simplified proof of the k-median local search algorithm of Arya et al. [3].

Definition 2 (very good, good, and bad facilities). *We call a facility $s \in S$* very good, *if $\eta^{-1}(s) = \emptyset$;* good, *if $\eta^{-1}(s) \neq \emptyset$ and no facility in $\eta^{-1}(s)$ has the same color as s; and* bad, *if some facility in $\eta^{-1}(s)$ has the same color as s.*

2.1 The Blocks

We now present a procedure (see Figure 1) to partition the set R^* into $R_1^*, \ldots, R_t^*$, the set B^* into $B_1^*, \ldots, B_t^*$, the set R into $R_1, \ldots, R_t$, and the set B into $B_1, \ldots, B_t$ for some integer t. The parts R_i^*, B_i^*, R_i, B_i are said to form `block-`i for $i = 1, \ldots, t$. Note that this procedure is used only for the sake of analysis. It shows how to first compute `block-1` and then recursively compute `block-`i for $i = 2, \ldots, t$.

Lemma 1. *The partitions of R^*, B^*, R, and B computed in Figure 1 satisfy the following properties.*

Compute `block`-1, i.e., R_1^*, B_1^*, R_1, and B_1:

1. Start with $R_1^* = B_1^* = R_1 = B_1 = \emptyset$.
2. If there is a bad facility $r \in R$ such that $|\eta^{-1}(r)| = 1$, then let $R_1 = \{r\}$, $B_1 = \emptyset$, $R_1^* = \eta^{-1}(r)$, $B_1^* = \emptyset$, and stop. If there is a bad facility $b \in B$ such that $|\eta^{-1}(b)| = 1$, then let $R_1 = \emptyset$, $B_1 = \{b\}$, $R_1^* = \emptyset$, $B_1^* = \eta^{-1}(b)$, and stop.
3. If there are good facilities $r \in R$ and $b \in B$ such that $|\eta^{-1}(r)| = |\eta^{-1}(b)| = 1$, then let $R_1 = \{r\}$, $B_1 = \{b\}$, $R_1^* = \eta^{-1}(b)$, $B_1^* = \eta^{-1}(r)$, and stop.
4. If there is no bad facility in S, let $R_1^* = R^*$, $B_1^* = B^*$, $R_1 = R$, $B_1 = B$, and stop. Otherwise let $s \in S$ be a bad facility such that $|\eta^{-1}(s)|$ is maximum.
5. Add s to either R_1 or B_1 according to whether it is a red or a blue facility. Add facilities in $\eta^{-1}(s)$ to R_1^* and B_1^* according to their color. If $|R_1^*| = |R_1|$ and $|B_1^*| = |B_1|$, then stop.
6. If $|R_1^*| > |R_1|$, then add $|R_1^*| - |R_1|$ very good or good red facilities to R_1. While doing so, give a preference to very good red facilities. For each facility s thus added to R_1, add facilities in $\eta^{-1}(s)$ to B_1^*.
7. If on the other hand $|B_1^*| > |B_1|$, then add $|B_1^*| - |B_1|$ very good or good blue facilities to B_1. While doing so, give a preference to very good blue facilities. For each facility s thus added to B_1, add facilities in $\eta^{-1}(s)$ to R_1^*.
8. Repeat steps 6 and 7 until we have $|R_1^*| = |R_1|$ and $|B_1^*| = |B_1|$, and then stop.

Recurse on $R^* \setminus R_1^*$, $B^* \setminus B_1^*$, $R \setminus R_1$, and $B \setminus B_1$ to compute `block`-i for $i \geq 2$.

Fig. 1. A procedure to compute partitions of R^*, B^*, R, and B

1. $|R_i^*| = |R_i|$ *and* $|B_i^*| = |B_i|$ *for all* $i = 1, \ldots, t$.
2. *For each* $o \in R_i^* \cup B_i^*$, *we have* $\eta(o) \in R_i \cup B_i$ *for* $i = 1, \ldots, t$.
3. *For* $i = 1, \ldots, t$, *at most one facility in* $R_i \cup B_i$ *is bad. We call such a facility* leader.
4. *For* $i = 1, \ldots, t$, *if there is a leader in* $R_i \cup B_i$, *we have*
   (a) *either all facilities in* R_i, *except the leader, are very good,*
   (b) *or all facilities in* B_i, *except the leader, are very good.*

The proof of this lemma is omitted due to lack of space.

2.2 The Swaps

Since $S = R \cup B$ is a local solution, any swap of a red facility and a blue facility does not decrease the cost of the solution, i.e., $\mathbf{cost}(R - r^- + r^+, B - b^- + b^+) \geq \mathbf{cost}(R, B)$. We use $\mathtt{swap}(r^-, r^+ \mid b^-, b^+)$ to denote this swap. When $r^- = r^+$, we also use $\mathtt{swap}(b^-, b^+)$ to denote this swap. Similarly, when $b^- = b^+$, we also use $\mathtt{swap}(r^-, r^+)$ to denote this swap. We now consider several inequalities of this type and add them to get the desired result. For each such swap considered below, we upper bound $\mathbf{cost}(R - r^- + r^+, B - b^- + b^+) - \mathbf{cost}(R, B)$ by giving a feasible assignment of clients to facilities.

Recall the definition of valid swaps; we call a swap valid if it does not change the number of red and blue facilities in the solution.

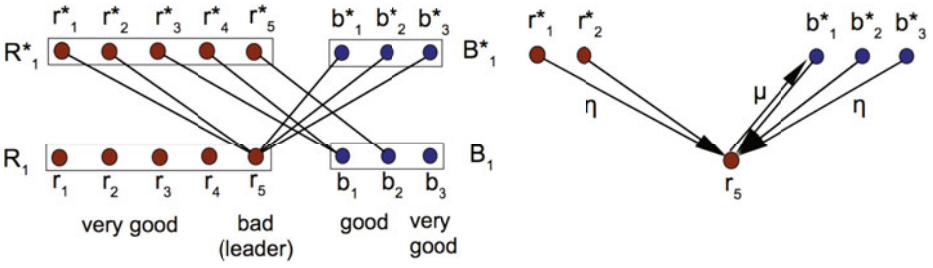

Fig. 2. On the left is an example of `block`-1: the facilities $R_1^*, B_1^* \subset O$ are shown at the top while $R_1, B_1 \subset S$ are shown at the bottom. We draw an edge between each $o \in R_1^* \cup B_1^*$ and $\eta(o) \in R_1 \cup B_1$. A single bad facility in $R_1 \cup B_1$, called leader, is r_5. The facilities $r_1, \ldots, r_4, b_3$ are very good while the facilities b_1, b_2 are good. Here case 4(a) holds. On the right is an example of functions η and μ. We have $\mu(r_5) = b_1^* \in \eta^{-1}(r_5)$.

Lemma 2. *1. Let $s \in R_1$ (resp. $s \in B_1$) be a very good facility and $o \in R_1^*$ (resp. $o \in B_1^*$) be any facility. Then*

$$\sum_{j \in N^*(o)} (O_j - S_j) + \sum_{j \in N(s) \setminus N^*(o)} 2O_j \geq 0. \tag{1}$$

2. Let $s \in R_1$ (resp. $s \in B_1$) be either good or bad facility such that $o = \mu(s) \in R_1^*$ (resp. $o \in B_1^*$). Then

$$\sum_{j \in N^*(o)} (O_j - S_j) + \sum_{j \in N(s) \cap N^*(\eta^{-1}(s) \setminus \{o\})} (O_j + S_j) + \sum_{j \in N(s) \setminus N^*(\eta^{-1}(s))} 2O_j \geq 0. \tag{2}$$

3. Let $s_1 \in R_1 \cup B_1$ be either good or bad facility, $s_2 \in R_1 \cup B_1$ be a very good facility, and $o_2 \in R_1^* \cup B_1^*$ be any facility such that deleting s_1, s_2 and adding $o_1 = \mu(s_1), o_2$ is a valid swap. Then

$$\sum_{\substack{j \in N^*(o_1) \\ \cup N^*(o_2)}} (O_j - S_j) + \sum_{\substack{j \in [N(s_1) \cup N(s_2)] \cap \\ [N^*(\eta^{-1}(s_1) \setminus \{o_1, o_2\})]}} (3O_j + S_j) + \sum_{\substack{j \in [N(s_1) \cup N(s_2)] \setminus \\ [N^*(\eta^{-1}(s_1) \cup \{o_2\})]}} 2O_j \geq 0. \tag{3}$$

4. Let $s_1, s_2 \in R_1 \cup B_1$ be either good or bad facilities such that deleting s_1, s_2 and adding $o_1 = \mu(s_1), o_2 = \mu(s_2)$ is a valid swap. Then

$$\sum_{\substack{j \in N^*(o_1) \\ \cup N^*(o_2)}} (O_j - S_j) + \sum_{\substack{j \in [N(s_1) \cup N(s_2)] \cap \\ [N^*(\eta^{-1}(s_1) \setminus \{o_1\}) \cup N^*(\eta^{-1}(s_2) \setminus \{o_2\})]}} (3O_j + S_j) + \sum_{\substack{j \in [N(s_1) \cup N(s_2)] \setminus \\ [N^*(\eta^{-1}(s_1)) \cup N^*(\eta^{-1}(s_2))]}} 2O_j \geq 0. \tag{4}$$

Proof. For a client j, let $s(j)$ denote the facility that serves j in solution S and let $o(j)$ denote the facility that serves j in solution O.

For item 1, consider $\texttt{swap}(s, o)$. We reroute clients as follows. A client $j \in N^*(o)$ is rerouted to o and thus the increase in its service cost is $O_j - S_j$. A client $j \in N(s) \setminus N^*(o)$ is rerouted to $\eta(o(j))$. Note that $\eta(o(j)) \neq s$ since s

is very good. The increase in its service cost is thus $d(j, \eta(o(j))) - S_j \leq O_j + d(o(j), \eta(o(j))) - S_j \leq O_j + d(o(j), s(j)) - S_j \leq O_j + O_j + S_j - S_j = 2O_j$. This sequence of inequalities follows from repeated use of triangle inequality. The clients not in $N^*(o) \cup N(s)$ are not rerouted. This proves item 1.

For item 2, consider $\texttt{swap}(s, o)$. A client $j \in N^*(o)$ is rerouted to o and thus the increase in its service cost is $O_j - S_j$. Consider a client $j \in N(s) \setminus N^*(\eta^{-1}(s))$. Since $o(j) \notin \eta^{-1}(s)$, we have $\eta(o(j)) \neq s$. Such a client is therefore rerouted to $\eta(o(j))$ and thus the increase in its service cost is $d(j, \eta(o(j))) - S_j \leq 2O_j$ as shown in item 1. A client $j \in N(s) \cap N^*(\eta^{-1}(s) \setminus \{o\})$ is rerouted to o and thus the increase in its service cost is $d(j, o) - S_j \leq d(j, s(j)) + d(s(j), o) - S_j \leq S_j + d(s(j), o(j)) - S_j \leq O_j + S_j$. Here $d(s(j), o) \leq d(s(j), o(j))$ follows from $o(j) \in \eta^{-1}(s)$, $o = \mu(s)$, and the definition of μ. The clients not in $N^*(o) \cup N(s)$ are not rerouted. This proves item 2.

The proofs of items 3 and 4 are very similar. Therefore we prove item 4 and omit the proof of item 3. For item 4, consider the swap: delete s_1, s_2 and add o_1, o_2. In this swap, we reroute the clients as follows. A client $j \in N^*(o_1)$ is rerouted to o_1 and a client $j \in N^*(o_2)$ is rerouted to o_2. Clearly the increase in service cost of clients $j \in N^*(o_1) \cup N^*(o_2)$ is $O_j - S_j$.

Now consider a client $j \in [N(s_1) \cup N(s_2)] \setminus [N^*(o_1) \cup N^*(o_2)]$. Assume without loss of generality that $j \in N(s_1)$; a similar argument also holds for the case $j \in N(s_2)$. Let $o(j)$ be the facility that serves j in O. If $\eta(o(j)) = s_1$, then j is rerouted to o_1 and the increase in service cost is $d(j, o_1) - d(j, s_1) \leq d(s_1, o_1) \leq d(s_1, o(j)) \leq S_j + O_j$. This sequence of inequalities follows from repeated use of triangle inequality and from the fact $o_1 = \mu(s_1)$. If $\eta(o(j)) = s_2$, then it is rerouted to o_2 and the increase in service cost is $d(j, o_2) - S_j \leq d(j, o(j)) + d(o(j), s_2) + d(s_2, o_2) - S_j \leq d(j, o(j)) + d(o(j), s_2) + d(s_2, o(j)) - S_j \leq d(j, o(j)) + d(o(j), s_1) + d(s_1, o(j)) - S_j \leq O_j + 2(O_j + S_j) - S_j = 3O_j + S_j$. This sequence of inequalities follows from repeated use of triangle inequality and from the fact $o_2 = \mu(s_2)$ and $\eta(o(j)) = s_2$. Now consider the case that $\eta(o(j))$ is neither s_1 or s_2. Let $s(j)$ denote the facility that serves j in S. We reroute j to $\eta(o(j))$ and the increase in service cost is thus $d(j, \eta(o(j))) - S_j \leq O_j + d(o(j), \eta(o(j))) - S_j \leq O_j + d(o(j), s(j)) - S_j \leq O_j + O_j + S_j - S_j = 2O_j$. This proves item 4.

2.3 Putting Together

Intuition. Note that inequality (1) has "$-S_j$" terms for some clients and "$+O_j$" terms for some clients. The analysis of Arya et al. [3] or Gupta and Tangwongsan [14] is based on adding several inequalities of this type so that the "$-S_j$" term is included for each client j once and "$+O_j$" term is included for each client j at most 5 times. Thus overall, they get $-\sum_j S_j + 5\sum_j O_j \geq 0$. This directly gives a 5-approximation. Unfortunately, such an analysis does not work in our setting. We also have to add several inequalities (2)-(4), thus incurring "$+S_j$" terms for some clients. We then use inequality (1) repeatedly to "cancel" the "$+S_j$" terms in order to prove a constant approximation. All the swaps

to be considered are contained in a block. For block-i, we prove the following inequality:

$$\sum_{j \in N^*(R_i^* \cup B_i^*)} S_j \leq O(1) \cdot \left[\sum_{j \in N^*(R_i^* \cup B_i^*)} O_j + \sum_{j \in N(R_i \cup B_i)} O_j \right]. \tag{5}$$

Adding these inequalities over all the blocks, we get a constant approximation:

$$\begin{aligned} \mathbf{cost}(S) &= \sum_{i=1}^{t} \sum_{j \in N^*(R_i^* \cup B_i^*)} S_j \leq O(1) \cdot \sum_{i=1}^{t} \left[\sum_{j \in N^*(R_i^* \cup B_i^*)} O_j + \sum_{j \in N(R_i \cup B_i)} O_j \right] \\ &\leq O(1) \cdot 2 \cdot \mathbf{cost}(O). \end{aligned}$$

The proof of inequality (5) is omitted due to lack of space. However here we illustrate how to prove it using the example in Figure 2.

We start with some notation. If R_1 has at least one good or very good facility, we fix a function $\mathtt{g} : R_1^* \to R_1$ such that each facility in $\mathtt{g}(R_1^*)$ is either good or very good and $|\mathtt{g}^{-1}(r)| \leq 2$ for all $r \in R_1$. It is easy to see that such a function exists. Similarly, if B_1 has at least one good or very good facility, we fix a function $\mathtt{g} : B_1^* \to B_1$ such that each facility in $\mathtt{g}(B_1^*)$ is either good or very good and $|\mathtt{g}^{-1}(b)| \leq 2$ for all $b \in B_1$.

Example in Figure 2. To convey our intuition, we prove inequality (5) for the example of `block`-1 in Figure 2. For concreteness, assume that the function μ is given by $r_5 \mapsto b_1^*, b_1 \mapsto r_4^*, b_2 \mapsto r_5^*$. Also assume that $\mathtt{g}$ is given by $r_1^* \mapsto r_1, r_2^* \mapsto r_2, r_3^* \mapsto r_3, r_4^* \mapsto r_4, r_5^* \mapsto r_4, b_1^* \mapsto b_1, b_2^* \mapsto b_2, b_3^* \mapsto b_3$. To obtain "$-S_j$" terms for clients in $N^*(B_1^*)$, we consider the following swaps and the corresponding inequalities:

- $\mathtt{swap}(\mathtt{g}(\mu(\mathtt{g}(b_1^*))), \mu(\mathtt{g}(b_1^*)) \mid \mathtt{g}(b_1^*), b_1^*)$ which is same as $\mathtt{swap}(r_4, r_4^* \mid b_1, b_1^*)$ (consider inequality (3)),
- $\mathtt{swap}(\mathtt{g}(\mu(\mathtt{g}(b_2^*))), \mu(\mathtt{g}(b_2^*)) \mid \mathtt{g}(b_2^*), b_2^*)$ which is same as $\mathtt{swap}(r_4, r_5^* \mid b_2, b_2^*)$ (consider inequality (3)),
- $\mathtt{swap}(\mathtt{g}(b_3^*), b_3^*)$ which is same as $\mathtt{swap}(b_3, b_3^*)$ (consider inequality (1)).

If we add these three inequalities, we get

$$\sum_{j \in N^*(\{r_4^*, b_1^*, r_5^*, b_2^*, b_3^*\})} (O_j - S_j) + \sum_{j \in N^*(r_3^*)} (3O_j + S_j) + \sum_{j \in N(\{b_3, r_4, b_1\})} 2O_j + \sum_{j \in N(\{r_4, b_2\})} 2O_j \geq 0. \tag{6}$$

We next consider the following the following swaps and the corresponding inequalities:

- $\mathtt{swap}(\mathtt{g}(r_1^*), r_1^*)$ which is same as $\mathtt{swap}(r_1, r_1^*)$ (consider inequality (1)),
- $\mathtt{swap}(\mathtt{g}(r_2^*), r_2^*)$ which is same as $\mathtt{swap}(r_2, r_2^*)$ (consider inequality (1)),
- $\mathtt{swap}(\mathtt{g}(r_3^*), r_3^*)$ which is same as $\mathtt{swap}(r_3, r_3^*)$ (consider inequality (1)). We in fact multiply this inequality by factor 2 in order to cancel the "$+S_j$" term in the second term of (6) above.

Adding these three inequalities, we get

$$\sum_{j \in N^*(\{r_1^*, r_2^*\})} (O_j - S_j) + 2 \sum_{j \in N^*(r_3^*)} (O_j - S_j) + \sum_{j \in N(\{r_1, r_2\})} 2O_j + 2 \sum_{j \in N(r_3)} 2O_j \geq 0. \quad (7)$$

Adding (6) and (7), we get our desired inequality

$$\sum_{j \in N^*(R_1^* \cup B_1^*)} S_j \leq 5 \sum_{j \in N^*(R_1^* \cup B_1^*)} O_j + 4 \sum_{j \in N(R_1 \cup B_1)} O_j.$$

3 Proof of Theorem 2

In this section, we outline the proof of Theorem 2. We consider the multi-swap local search algorithm of Arya et al. [3]: start with any k facilities in the solution S and output a local optimum solution w.r.t. the following q-swap operation: delete q facilities from S and add q facilities in $\mathcal{F} \setminus S$ to S. We use a notation similar to the previous section. In addition, let $P \subseteq \mathcal{C}$ denote the set of clients that pay penalty in the locally optimum solution S and let $P^* \subseteq \mathcal{C}$ denote the set of clients that pay penalty in the optimum solution O. We prove the following theorem which implies that S is a $(3 + 2/q)$-approximation.

Theorem 4

$$\sum_{j \notin P} S_j + \sum_{j \in P} p_j \leq \left(3 + \frac{2}{q}\right) \sum_{j \notin P^*} O_j + \left(1 + \frac{1}{q}\right) \sum_{j \in P^*} p_j.$$

Note that even if the multiplier of $\sum_{j \in P^*} p_j$ on the right is $(1 + 1/q)$ instead of 1, one may use the above result, as a subroutine, in the algorithm for the robust k-median problem [8]. This is a version of the k-median problem in which at most l clients may be left unserved. We obtain a solution which has number of outliers at most $l(1 + \epsilon)(1 + \gamma)$ and has cost at most $(3 + \epsilon)(1 + 1/\gamma)$ for any fixed $\epsilon, \gamma > 0$. We omit further details from here.

The proof of Theorem 4 is omitted due to lack of space.

References

1. Archer, A., Rajagopalan, R., Shmoys, D.B.: Lagrangian relaxation for the k-median problem: New insights and continuity properties. In: Di Battista, G., Zwick, U. (eds.) ESA 2003. LNCS, vol. 2832, pp. 31–42. Springer, Heidelberg (2003)
2. Arora, S., Raghavan, P., Rao, S.: Approximation schemes for Euclidean k-medians and related problems. In: STOC 1998 (1998)
3. Arya, V., Garg, N., Khandekar, R., Meyerson, A., Munagala, K., Pandit, V.: Local search heuristics for k-median and facility location problems. SIAM J. Comput. 33(3), 544–562 (2004)
4. Bartal, Y.: On approximating arbitrary metrices by tree metrics. In: STOC 1998 (1998)

5. Bateni, M., Hajiaghayi, M.: Assignment problem in content distribution networks: unsplittable hard-capacitated facility location. In: SODA 2009 (2009)
6. Charikar, M., Guha, S.: Improved combinatorial algorithms for facility location problems. SIAM J. Comput. 34(4), 803–824 (2005)
7. Charikar, M., Guha, S., Tardos, É., Shmoys, D.: A constant-factor approximation algorithm for the k-median problem. J. Comp. Sys. Sci. 65(1), 129–149 (2002)
8. Charikar, M., Khuller, S., Mount, D.M., Narasimhan, G.: Algorithms for facility location problems with outliers. In: SODA 2001 (2001)
9. Chudak, F.A., Williamson, D.P.: Improved approximation algorithms for capacitated facility location problems. Math. Program. 102(2), 207–222 (2005)
10. Devanur, N.R., Garg, N., Khandekar, R., Pandit, V., Rohit, K., Vinayaka, P., Saberi, A., Vazirani, V.V.: Price of anarchy, locality gap, and a network service provider game. In: Deng, X., Ye, Y. (eds.) WINE 2005. LNCS, vol. 3828, pp. 1046–1055. Springer, Heidelberg (2005)
11. Fakcharoenphol, J., Rao, S., Talwar, K.: A tight bound on approximating arbitrary metrics by tree metrics. J. Comput. System Sci. 69(3), 485–497 (2004)
12. Feige, U., Mirrokni, V.S., Vondrak, J.: Maximizing non-monotone submodular functions. In: FOCS 2007 (2007)
13. Fleischer, L., Goemans, M.X., Mirrokni, V.S., Sviridenko, M.: Tight approximation algorithms for maximum general assignment problems. In: SODA 2006 (2006)
14. Gupta, A., Tangwongsan, K.: Simpler analyses of local search algorithms for facility location. ArXiv e-prints, arXiv:0809.2554 (2008)
15. Jain, K.: Private communication (2009)
16. Jain, K., Mahdian, M., Saberi, A.: A new greedy approach for facility location problems. In: STOC 2002 (2002)
17. Jain, K., Vazirani, V.V.: Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and Lagrangian relaxation. Journal of the ACM 48(2), 274–296 (2001)
18. Kolliopoulos, S.G., Rao, S.: A nearly linear-time approximation scheme for the Euclidean k-median problem. SIAM J. Comput. 37(3), 757–782 (2007)
19. Korupolu, M.R., Plaxton, C.G., Rajaraman, R.: Analysis of a local search heuristic for facility location problems. Journal of Algorithms 37(1), 146–188 (2000)
20. Kuehn, A., Hamburger, M.: A heuristic program for locating warehouses. Management Science 9, 643–666 (1963)
21. Lin, J.-H., Vitter, J.S.: Approximation algorithms for geometric median problems. Inf. Process. Lett. 44(5), 245–249 (1992)
22. Mahdian, M., Pál, M.: Universal facility location. In: Di Battista, G., Zwick, U. (eds.) ESA 2003. LNCS, vol. 2832, pp. 409–421. Springer, Heidelberg (2003)
23. Megiddo, N., Supowit, K.J.: On the complexity of some common geometric location problems. SIAM J. Comput. 13(1), 182–196 (1984)
24. Pál, M., Tardos, É., Wexler, T.: Facility location with nonuniform hard capacities. In: FOCS 2001 (2001)
25. Zhang, J., Chen, B., Ye, Y.: A multi-exchange local search algorithm for the capacitated facility location problem. In: Bienstock, D., Nemhauser, G.L. (eds.) IPCO 2004. LNCS, vol. 3064, pp. 219–233. Springer, Heidelberg (2004)