

Locating Depots for Capacitated Vehicle Routing*

Inge Li Gørtz[†]

Viswanath Nagarajan[‡]

Abstract

We study a location-routing problem in the context of capacitated vehicle routing. The input to the k -location capacitated vehicle routing problem (k -LocVRP) consists of a set of demand locations in a metric space and a fleet of k identical vehicles, each of capacity Q . The objective is to locate k depots, one for each vehicle, and compute routes for the vehicles so that all demands are satisfied and the total cost is minimized. Our main result is a constant-factor approximation algorithm for k -LocVRP. In obtaining this result, we introduce a common generalization of the k -median and *minimum spanning tree* problems (called k median forest), which might be of independent interest. We give a local-search based $(3 + \epsilon)$ -approximation algorithm for k median forest, which leads to a $(12 + \epsilon)$ -approximation algorithm for k -LocVRP, for any constant $\epsilon > 0$.

Keywords: Vehicle routing, facility location, approximation algorithms.

1 Introduction

In typical facility location problems, one wishes to locate centers and connect clients directly to centers at minimum cost. On the other hand, the goal in vehicle routing problems (VRPs) is to compute routes for vehicles originating from a given set of depots. Location routing problems represent an integrated approach, where we wish to make combined decisions on facility location and vehicle routing. This is a widely researched area in operations research, see eg. surveys [4, 17, 18, 5, 21, 22]. Most of these papers deal with either exact methods or heuristics without any performance guarantees. In this paper we present an approximation algorithm for a location routing problem in the context of capacitated vehicle routing.

Capacitated vehicle routing (CVRP) is an extensively studied vehicle routing problem [24] which involves distributing units of an identical item to a set of demand locations. Formally, we are given a metric space (V, d) on vertices V with distance function $d : V \times V \rightarrow \mathbb{R}_+$ that is symmetric and satisfies triangle inequality. Each vertex $u \in V$ demands q_u units of the item. We have available a fleet of k vehicles, each having capacity Q and located at specified depots. The goal is to distribute items using the k vehicles at minimum total cost. There are two versions of CVRP depending on whether or not the demand at a vertex may be satisfied over multiple visits. We focus on the *unsplit delivery* version in the paper, while noting that this also implies the result under split-deliveries.

We consider the question “where should one locate the k depots so that the resulting vehicle routing solution has minimum cost?” This is called *k -location capacitated vehicle routing (k -LocVRP)*.

*A preliminary version appeared in the proceedings of the Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX) 2011.

[†]Technical University of Denmark. Supported by the Danish Council for Independent Research | Natural Sciences.

[‡]IOE Department, University of Michigan.

This is the author manuscript accepted for publication and has undergone full peer review but has not been through the copyediting, typesetting, pagination and proofreading process, which may lead to differences between this version and the [Version record](#). Please cite this article as [doi:10.1002/net.21683](https://doi.org/10.1002/net.21683).

The k -LocVRP problem bears obvious similarity to the well-known k median problem, where the goal is to choose k centers to minimize the sum of distances of each vertex to its closest center. In fact, k -median is a special case of k -LocVRP when the vehicle capacity $Q = 1$. Not surprisingly, our algorithm for k -LocVRP builds on approximation algorithms for the k median problem. However k -LocVRP is more complex under general (finite) capacities since it needs to handle the routing aspect as well.

In obtaining an algorithm for k -LocVRP we introduce a new location problem, k median forest, which might be of independent interest. The input here consists of a metric (V, d) , vertex-weights $\{q_u\}_{u \in V}$, bound k and parameter $\rho \in \mathbb{R}_+$. The goal is to find a subset $S \subseteq V$ with $|S| = k$ minimizing $\sum_{u \in V} q_u \cdot d(u, S) + \rho \cdot d(\text{MST}(V/S))$. Here $d(u, S) = \min_{w \in S} d(u, w)$ is the minimum distance between u and a vertex in S , and $d(\text{MST}(V/S))$ is the length of the minimum spanning tree in the graph obtained by contracting S to a single vertex. As a potential application, consider a telecommunication company that wants to locate k facilities so as to maintain two types of connectivity to its clients: (i) a low-cost cable that connects each client directly to its nearest facility, and (ii) a path consisting of high-cost cables that connects each client to some facility. When the cost-per-length of the high-cost cable is ρ times that of the low-cost cable, this corresponds precisely to the k median forest problem (with all vertex-weights $q_u = 1$).

Standard Definitions An algorithm for a minimization problem is said to have *approximation ratio* α (also called an α -approximation algorithm) if it runs in polynomial time and on every instance of the problem it produces a solution having objective at most α times the optimum; see the books [25, 26] for more background on approximation algorithms.

Given a local search algorithm for a minimization problem, a *locally optimal* solution L is one where no solution in the local neighborhood of L has smaller objective value than L . The *locality gap* of such an algorithm is defined to be the maximum over all problem instances I and locally optimal solutions L_I for I , of the ratio of L_I 's objective to the optimal value of I , i.e.

$$\text{Locality gap} = \max_{\text{instance } I} \max_{\text{local opt } L_I} \frac{\text{Objective}(L_I)}{\text{Optimum}(I)}.$$

If the local search algorithm terminates in polynomial time, its approximation ratio is upper bounded by the locality gap.

Our Results and Techniques The main result is the following.

Theorem 1 *There is a $(12 + \epsilon)$ -approximation algorithm for k -LocVRP, for any constant $\epsilon > 0$.*

Our algorithm first reduces k -LocVRP to k median forest, at the loss of an approximation factor of four. This step is fairly straightforward and makes use of known lower-bounds [12] for the CVRP problem. We present this reduction in Section 2. Then, in Section 3 we prove the following result which implies Theorem 1.

Theorem 2 *There is a $(3 + \epsilon)$ -approximation algorithm for k median forest, for any constant $\epsilon > 0$.*

This is the technically most interesting part of the paper. The algorithm is straightforward: perform local search using multi-swaps. It is well known that (single swap) local search is optimal for the minimum spanning tree problem. Moreover, Arya et al. [3] showed that t -swap local search achieves

exactly a $(3 + \frac{2}{t})$ -approximation ratio for the k -median objective (this proof was later simplified by Gupta and Tangwongsan [11]). Thus one can hope that local search performs well for k median forest, which is a combination of both MST and k -median objectives. However, the local moves used in proving the quality of local optima are different for the MST and k -median objectives. Our proof shows we can *simultaneously* bound both MST and k -median objectives using a common set of local moves. In fact we prove that the locality gap for k median forest under t -swaps is also $(3 + \frac{2}{t})$. To bound the k -median part of the objective due to these swaps, we use the result from [11]. The interesting part of the proof is in bounding the change in the MST cost due to these swaps—this makes use of non-trivial exchange properties of spanning trees (that are easier to describe in a matroid context [23]) and additional properties of the potential swaps from [11].

We note that the *uniformity* of metrics in the k -median and MST parts of the objective is crucial for the local search algorithm to perform well. We show (in Appendix B) that the locality gap is unbounded for the more general “non-uniform” k median forest problem (where there are different metrics for the k -median and MST parts). Still, we note that one can obtain a constant-factor approximation algorithm for non-uniform k median forest using a linear-programming based approach from [16] for the *matroid median problem*. We omit the description of this result here since it is not directly related to the location-routing problem; the interested reader is referred to [10].

Related Work The basic capacitated vehicle routing problem involves a single fixed depot. There are two versions of CVRP: *split delivery* where the demand of a vertex may be satisfied over multiple visits; and *unsplit delivery* where the demand at a vertex must be satisfied in a single visit (in this case we also assume $\max_{u \in V} q_u \leq Q$). Observe that the optimal value under split-delivery is at most that under unsplit-delivery. The best known approximation guarantee for split-delivery CVRP is $\alpha + 1$ [12, 2] and for unsplit-delivery CVRP is $\alpha + 2$ [1], where α denotes the best approximation ratio for the Traveling Salesman Problem. We make use of the following known lower bounds for CVRP with single depot r : the minimum TSP tour on all demand locations, and $\frac{2}{Q} \sum_{u \in V} d(r, u) \cdot q_u$. Constant factor approximation algorithms are also known for the CVRP with multiple (fixed) depots [19].

The k median problem is a widely studied location problem and has many constant factor approximation algorithms. Starting with the LP-rounding algorithm of [8], the primal-dual approach was used in [15], and also local search [3]. A simpler analysis of the local search algorithm was given in [11]; we make use of this in our proof for the k median forest problem. For a long time, the $(3 + \epsilon)$ -approximation algorithm via local search [3] was the best bound known for k -median. Very recently, this bound has been improved to $2.675 + \epsilon$ in [6, 20].

Recently [13] studied (among other problems) a facility-location variant of CVRP: there are opening costs for depots and the goal is to open a set of depots and find vehicle routes so as to minimize the sum of opening and routing costs. The relation between k -LocVRP and the problem in [13] is similar to that between k -median and facility location. In [13] the authors gave a 4.38-approximation algorithm for facility-location CVRP. Following a similar approach one can obtain a “bicriteria” approximation algorithm for k -LocVRP, where $2k$ depots are opened but the algorithm’s objective is compared to the optimum using only k depots. However more work is needed to obtain a true approximation, and this is precisely where we make use of the k median forest problem.

In independent work, Carnes and Shmoys [7] gave a 2-approximation algorithm for the k -location

routing problem, which is the uncapacitated version of k -LocVRP. Their algorithm is based on a primal-dual schema and Langrangian relaxation. The facility location variant of this problem was considered earlier by Goemans and Williamson [9], who gave a 2-approximation algorithm as part of their general framework for network design problems.

2 Reducing k -LocVRP to k Median Forest

We show that the k -LocVRP problem can be reduced to k median forest at the loss of a constant approximation factor. This makes use of known lower bounds for CVRP [12, 19, 13].

For any subset $S \subseteq V$, let $\text{Flow}(S) := \frac{2}{Q} \sum_{u \in V} q_u \cdot d(u, S)$, and let $\text{Tree}(S) = d(\text{MST}(V/S))$ be the length of the minimum spanning tree in the metric obtained by contracting S . The following theorem is implicit in previous work [12, 19, 13]; this uses a natural MST splitting algorithm.

Theorem 3 (Harks et al. [13]) *Given any instance of CVRP on metric (V, d) with demands $\{q_u\}_{u \in V}$, vehicle capacity Q and depots $S \subseteq V$,*

- *The optimal value of split-delivery CVRP is at least $\max\{\text{Flow}(S), \text{Tree}(S)\}$.*
- *There is a polynomial time algorithm that computes an unsplit-delivery solution of length at most $2 \cdot \text{Flow}(S) + 2 \cdot \text{Tree}(S)$.*

Based on this it is clear that, up to a factor of four, the optimal value of the CVRP instance given depot positions S equals $\text{Flow}(S) + \text{Tree}(S)$. The following lemma formalizes this reduction.

Lemma 4 *If there is a β -approximation algorithm for k median forest then there is a 4β -approximation algorithm for k -LocVRP.*

Proof: Let Opt denote the optimal value of the k -LocVRP instance. Using the lower bound in Theorem 3,

$$\text{Opt} \geq \min_{S:|S|=k} \max\{\text{Flow}(S), \text{Tree}(S)\} \geq \min_{S:|S|=k} [\epsilon \cdot \text{Flow}(S) + (1 - \epsilon) \cdot \text{Tree}(S)],$$

where $\epsilon \in [0, 1]$ is any value; this will be fixed later. Consider the instance of k median forest on metric (V, d) , vertex weights $\{q_u\}_{u \in V}$ and parameter $\rho = \frac{1-\epsilon}{\epsilon} \cdot \frac{Q}{2}$. For any $S \subseteq V$ the objective is:

$$\begin{aligned} \sum_{u \in V} q_u \cdot d(u, S) + \rho \cdot d(\text{MST}(V/S)) &= \frac{Q}{2} \cdot \text{Flow}(S) + \rho \cdot \text{Tree}(S) \\ &= \frac{Q}{2\epsilon} \cdot [\epsilon \cdot \text{Flow}(S) + (1 - \epsilon) \cdot \text{Tree}(S)]. \end{aligned}$$

Thus the optimal value of the k median forest instance is at most $\frac{Q}{2\epsilon} \cdot \text{Opt}$. Let S_{alg} denote the solution found by the β -approximation algorithm for k median forest. It follows that $|S_{alg}| = k$ and:

$$\epsilon \cdot \text{Flow}(S_{alg}) + (1 - \epsilon) \cdot \text{Tree}(S_{alg}) \leq \beta \cdot \text{Opt} \tag{1}$$

For the k -LocVRP instance, we locate the depots at S_{alg} . Using Theorem 3, the cost of the resulting vehicle routing solution is at most $2 \cdot \text{Flow}(S_{alg}) + 2 \cdot \text{Tree}(S_{alg}) = 4 \cdot [\epsilon \cdot \text{Flow}(S_{alg}) + (1 - \epsilon) \cdot \text{Tree}(S_{alg})]$

where we set $\epsilon = 1/2$. From Inequality (1) it follows that our algorithm is a 4β -approximation algorithm for k -LocVRP. \blacksquare

We note below that this reduction already gives us a constant factor *bicriteria* approximation algorithm for k -LocVRP. An algorithm for k -LocVRP is said to be an (α, β) -bicriteria approximation (for values $\alpha, \beta \geq 1$) if, on every problem instance it produces a solution using at most $\beta \cdot k$ depots with objective value at most α times the optimum which uses at most k depots.

Let S_{med} denote an approximate solution to k -median on metric (V, d) with vertex-weights $\{q_u : u \in V\}$, which can be obtained by directly using a k -median algorithm [3]. Let S_{mst} denote the optimal solution to $\min_{S: |S| \leq k} d(MST(V/S))$, which can be obtained using the greedy MST algorithm. We output $S_{bi} = S_{med} \cup S_{mst}$ as a solution to k -LocVRP, along with the vehicle routes obtained from Theorem 3 applied to S_{bi} . Note that $|S_{bi}| \leq 2k$, so we open at most $2k$ depots. Moreover, if S^* denotes the location of depots in the optimal solution to k -LocVRP then:

- $\text{Flow}(S_{med}) \leq (3 + \delta) \cdot \text{Flow}(S^*)$ since we used a $(3 + \delta)$ -approximation algorithm for k -median [3], for any constant $\delta > 0$.
- $\text{Tree}(S_{mst}) \leq \text{Tree}(S^*)$ since S_{mst} is an optimal solution to the MST part of the objective.

Clearly $\text{Flow}(S_{bi}) \leq \text{Flow}(S_{med})$ and $\text{Tree}(S_{bi}) \leq \text{Tree}(S_{mst})$, so:

$$\frac{1}{2} \cdot \text{Flow}(S_{bi}) + \frac{1}{2} \cdot \text{Tree}(S_{bi}) \leq \frac{3 + \delta}{2} \cdot [\text{Flow}(S^*) + \text{Tree}(S^*)] \leq (3 + \delta) \cdot \text{Opt}$$

Using Theorem 3 the cost of the CVRP solution with depots S_{bi} is at most $4(3 + \delta) \cdot \text{Opt}$. So this gives a $(12 + \delta, 2)$ bicriteria approximation algorithm for k -LocVRP, where $\delta > 0$ is any fixed constant. We note that this approach combined with algorithms for facility-location and Steiner tree immediately gives a constant factor approximation for the facility location CVRP considered in [13]; however this yields a worse constant factor. For k -LocVRP this approach clearly does not give any true approximation ratio, and to achieve this we give an algorithm for k median forest.

3 The k Median Forest Problem

The input to *k median forest* consists of a metric (V, d) , vertex-weights $\{q_u\}_{u \in V}$, bound k and parameter $\rho \in \mathbb{R}_+$. The goal is to find $S \subseteq V$ with $|S| = k$ minimizing:

$$\Phi(S) = \sum_{u \in V} q_u \cdot d(u, S) + \rho \cdot d(MST(V/S)),$$

where $d(u, S) = \min_{w \in S} d(u, w)$ and $MST(V/S)$ is the minimum spanning tree in the graph obtained by contracting S to a single vertex.

Note that when $\rho = 0$, we recover the *k -median* problem. And when ρ is very large, we obtain the *k -tree* problem where the goal is to find a minimum cost forest consisting of k trees that spans all the vertices.¹ Observe that the *k -tree* problem can be solved optimally using the greedy algorithm for minimum-spanning-tree and stopping when there are k components. Although the *k*

¹Equivalently, the *k -tree* problem involves choosing k centers $S \subseteq V$ so as to minimize $d(MST(V/S))$.

median forest objective is a linear combination of the k -median and k -tree objectives, we note (in Appendix A) that approximate solutions to these three problems can be very different from each other.

We analyze the natural t -swap local search for this problem, for any constant t . Starting at an arbitrary solution $L \subseteq V$ consisting of k centers, do the following until no improvement is possible: if there exists $D \subseteq L$ and $A \subseteq V \setminus L$ with $|D| = |A| \leq t$ and $\Phi((L \setminus D) \cup A) < \Phi(L)$ then set $L \leftarrow (L \setminus D) \cup A$. We prove that the locality gap of this procedure for k -median forest is at most $3 + \frac{2}{t}$. This is also tight since a matching lower bound of $3 + \frac{2}{t}$ is already known, even in the special case of k -median [3]. Somewhat surprisingly, it suffices to consider exactly the same set of swaps from [11] (for k -median) to establish our result, although these swaps did not take into account any MST contribution.

Running Time Clearly, each local step can be performed in $n^{O(t)}$ time where $n = |V|$. This is polynomial for fixed t . But the number of iterations to reach a local optimum may be super-polynomial. However this can be made polynomial by the standard method of performing a local move only if the cost Φ reduces by some $1 + \frac{1}{p(n)}$ factor where $p(n)$ is some polynomial in n (see eg. [3]). Under this modified local search, any local optimum $F \subseteq V$ satisfies the weaker condition:

$$\Phi((F \setminus D) \cup A) \geq \left(1 - \frac{1}{p(n)}\right) \cdot \Phi(F), \quad \forall D \subseteq F, A \subseteq V \setminus F, |D| = |A| \leq t.$$

To reduce notation, we present the analysis below for the original local search, i.e. no neighbor $F \setminus D \cup A$ of the local optimum F has cost less than $\Phi(F)$. Exactly the same analysis holds for the modified local search as well: the approximation factor only increases by an additive $o(1)$ term since the number of local inequalities used in the analysis will be bounded by $2n \ll p(n)$.

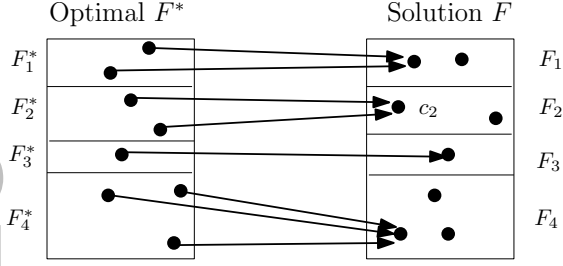
Let $F \subseteq V$ denote any local optimum solution (under t -swaps) and $F^* \subseteq V$ the global optimum. Note that $|F| = |F^*| = k$. Define map $\eta : F^* \rightarrow F$ as $\eta(w) = \arg \min_{v \in F} d(w, v)$ for all $w \in F^*$, i.e., for each optimum center $w \in F^*$, $\eta(w)$ is w 's closest center in F . For any $S \subseteq V$, define $\text{Med}(S) := \sum_{u \in V} q_u \cdot d(u, S)$, and let $\text{Tree}(S) = d(\text{MST}(V/S))$ be the length of the minimum spanning tree after contracting S to a single vertex; so $\Phi(S) = \text{Med}(S) + \rho \cdot \text{Tree}(S)$. For any $D \subseteq F$ and $A \subseteq V \setminus F$ with $|D| = |A| \leq t$ we use $F - D + A$ to denote $(F \setminus D) \cup A$ and refer to the local move from F to $F - D + A$ as a “ (D, A) swap”. We use the following swap construction from [11] for the k -median problem. Below, for any integer h , we use $[h] := \{1, 2, \dots, h\}$.

Theorem 5 (Gupta and Tangwongsan [11]) *For any $F, F^* \subseteq V$ with $|F| = |F^*| = k$, there are partitions $\{F_i\}_{i=1}^{\ell}$ of F and $\{F_i^*\}_{i=1}^{\ell}$ of F^* such that $\forall i \in [\ell]$, $|F_i| = |F_i^*|$. Furthermore, for each $i \in [\ell]$, there is a unique $c_i \in F_i$ such that $\eta(w) = c_i$ for all $w \in F_i^*$ and $\eta^{-1}(v) = \emptyset$ for all $v \in F_i \setminus \{c_i\}$. Define set \mathcal{S} of t -swaps with multipliers $\{\alpha(s) : s \in \mathcal{S}\}$ as:*

- For any $i \in [\ell]$, if $|F_i| \leq t$ then swap $(F_i, F_i^*) \in \mathcal{S}$ with $\alpha(F_i, F_i^*) = 1$.
- For any $i \in [\ell]$, if $|F_i| > t$ then for each $a \in F_i^*$ and $b \in F_i \setminus \{c_i\}$ swap $(b, a) \in \mathcal{S}$ with $\alpha(b, a) = \frac{1}{|F_i| - 1}$.

Then we have:

- $\sum_{(D,A) \in \mathcal{S}} \alpha(D, A) \cdot (\text{Med}(F - D + A) - \text{Med}(F)) \leq (3 + 2/t) \cdot \text{Med}(F^*) - \text{Med}(F)$.



Example has $\ell = 4$ parts.

Figure 1: The map $\eta : F^* \rightarrow F$ and partitions of F^* and F in the swaps of Theorem 5.

- For each $w \in F^*$, the extent to which w is added $\sum_{(D,A) \in \mathcal{S}: w \in A} \alpha(D, A) = 1$.
- For each $v \in F$, the extent to which v is dropped $\sum_{(D,A) \in \mathcal{S}: v \in D} \alpha(D, A) \leq 1 + \frac{1}{t}$.

See Figure 1 for an example.

We use the same set \mathcal{S} of swaps for the k median forest problem and will show the following:

$$\sum_{(D,A) \in \mathcal{S}} \alpha(D, A) \cdot (\text{Tree}(F - D + A) - \text{Tree}(F)) \leq (3 + 2/t) \cdot \text{Tree}(F^*) - \text{Tree}(F) \quad (2)$$

Multiplying this by ρ and adding it to the similar inequality in Theorem 5 for Med (since both inequalities use exactly the same set \mathcal{S} of swaps with respective multipliers) we obtain:

$$\sum_{(D,A) \in \mathcal{S}} \alpha(D, A) \cdot (\Phi(F - D + A) - \Phi(F)) \leq (3 + 2/t) \cdot \Phi(F^*) - \Phi(F). \quad (3)$$

Finally by local optimality of F , the left-hand-side above is non-negative, and we have:

Theorem 6 *The t -swap local search algorithm for k median forest achieves an approximation ratio of $(3 + \frac{2}{t})$.*

As noted above, the modified local search for polynomial running time has an approximation ratio of $3 + \frac{2}{t} + o(1)$. This uses the fact that the total extent of swaps used in (3) is $\sum_{(D,A) \in \mathcal{S}} \alpha(D, A) \leq 2n$.

It remains to prove (2), which we do in the rest of the section. The main idea is to upper bound $\text{Tree}(F - D + A) - \text{Tree}(F)$ for each swap $(D, A) \in \mathcal{S}$ in terms of certain edges from $\text{MST}(V/F^*)$; this is formalized in Lemma 13. This will make precise the connection between the swaps in Theorem 5 and the MST part of the objective Φ .

Before stating and proving the main lemma, we introduce some useful constructs and properties. By construction of the swaps \mathcal{S} in Theorem 5, observe that:

Claim 7 *For any swap $(D, A) \in \mathcal{S}$ and any $f^* \in F^* \setminus A$ we have $\eta(f^*) \in F \setminus D$.*

Augmenting the metric (V, d) In order to work with $\text{Tree}(S)$ for various subsets $S \subseteq V$, it is convenient to use an augmented graph H which is the complete graph on vertices $V \cup \{r\}$ where r is a new vertex. We let $E = \binom{V}{2}$ denote the edges in the metric. Then graph H has edges $E \cup \{(r, v) : v \in V\}$. The edges $\{(r, v) : v \in V\}$ are called *root-edges* and edges E are *true-edges*.

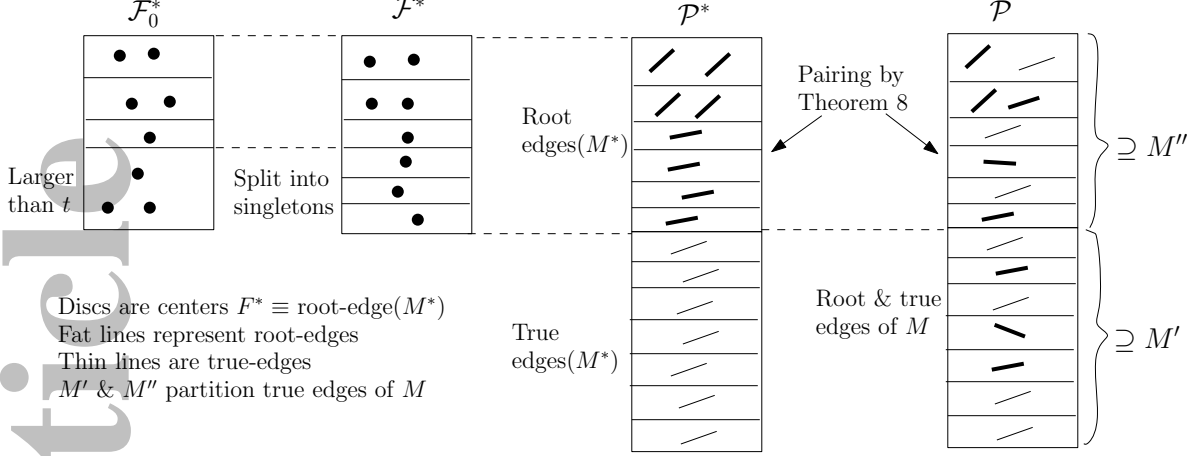


Figure 2: The partitions used in local search proof (eg. has $k = 8$ and $t = 2$).

For any subset $S \subseteq V$, note that $\text{Tree}(S)$ corresponds to the spanning tree in H containing root-edges $\{(r, v) : v \in S\}$ and true-edges $MST(V/S)$. For ease of notation, when it is clear from context, for subset $S \subseteq V$, we use S to also denote the set $\{(r, v) : v \in S\}$ of root-edges.

Let M denote the *spanning tree* of H consisting of edges $MST(V/F) \cup \{(r, v) : v \in F\}$. Similarly, M^* is the spanning tree $MST(V/F^*) \cup \{(r, v) : v \in F^*\}$. Note that M and M^* correspond to $\text{Tree}(F)$ and $\text{Tree}(F^*)$ respectively. An important step in proving (2) is to relate the edges of M and M^* to the swaps \mathcal{S} from Theorem 5: this is what enables charging the MST increases under different swaps in \mathcal{S} to M^* and M . This is done via the following exchange property of spanning trees (which holds more generally for any matroid), see Equation (42.15) in Schrijver [23].

Theorem 8 (Schrijver [23]) *Given two spanning trees T_1 and T_2 in a graph H and a partition $\{T_1(i)\}_{i=1}^p$ of the edges of T_1 , there exists a partition $\{T_2(i)\}_{i=1}^p$ of edges of T_2 such that $(T_2 \setminus T_2(i)) \cup T_1(i)$ is a spanning tree in H for each $i \in [p]$.*

Relating M , M^* and \mathcal{S} We will apply Theorem 8 on trees M^* and M and a partition of M^* derived from \mathcal{S} . Throughout, M^* and M represent the corresponding edge-sets. Recall the partition $\mathcal{F}_0^* := \{F_i^*\}_{i=1}^\ell$ of F^* from Theorem 5; we refine \mathcal{F}_0^* by splitting parts of size larger than t into singletons, and let \mathcal{F}^* denote the resulting partition (see Figure 2). The reason behind splitting the large parts of $\{F_i^*\}_{i=1}^\ell$ is to ensure the following property (recall the swaps \mathcal{S} from Theorem 5).

Claim 9 *For each swap $(D, A) \in \mathcal{S}$, $A \subseteq F^*$ appears as a part in \mathcal{F}^* . Moreover, for each part A' in \mathcal{F}^* there is some swap $(D', A') \in \mathcal{S}$.*

Consider the partition \mathcal{P}^* of the spanning tree M^* with parts $\mathcal{F}^* \cup \{e\}_{e \in M^* \setminus F^*}$, i.e., each true edge lies in a singleton part and the root edges form the partition \mathcal{F}^* defined above. Let \mathcal{P} denote the partition of spanning tree M obtained by applying Theorem 8 with partition \mathcal{P}^* of M^* ; note also that there is a *pairing* between parts of \mathcal{P} and \mathcal{P}^* . Let $M' \subseteq M \cap E$ denote the true edges of M that are paired with true edges of M^* ; and $M'' = (M \cap E) \setminus M'$ are the remaining true edges of M (which are paired with parts in \mathcal{F}^*). See Figure 2 for an example.

In the following analysis we will bound the cost of M' and M'' separately.

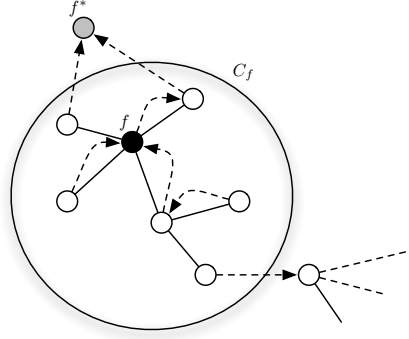


Figure 3: The component C_f . Solid edges are true edges in M , dashed arrows are the edges in T_f^* .

Claim 10 $\sum_{e \in M'} d_e \leq \sum_{h \in E \cap M^*} d_h$.

Proof: Fix any $e \in M'$. By the definition of M' it follows that there is a true-edge $h \in E \cap M^*$ such that part $\{h\}$ in \mathcal{P}^* is paired with part $\{e\}$ in \mathcal{P} . In particular, $M - e + h$ is a spanning tree in H . Note that the root edges in $M - e + h$ are exactly F , and so $M - e + h$ is a spanning tree in the original metric graph (V, E) when we contract vertices F . By definition, $M = MST(V/F)$ is the *minimum* spanning tree in the metric graph (V, E) when we contract vertices F . Hence we have $d(M) - d_e + d_h \geq d(M)$, implying $d_e \leq d_h$. Summing over all $e \in M'$ and observing that each edge $h \in E \cap M^*$ can be paired with at most one $e \in M'$, we obtain the claim. ■

By Claim 9 it follows that for each swap $(D, A) \in \mathcal{S}$, A is a part in \mathcal{F}^* (and so in \mathcal{P}^*). Let E_A be the (possibly empty) set of true-edges of M that are paired with the part A of \mathcal{P}^* .

Claim 11 $\{E_A : (D, A) \in \mathcal{S}\}$ is a partition of M'' .

Proof: Consider the partition \mathcal{P} of M given by Theorem 8 applied to \mathcal{P}^* . By definition, $M' \subseteq E \cap M$ are the true edges of M paired (by \mathcal{P} and \mathcal{P}^*) with true edges of M^* ; and $M'' = (E \cap M) \setminus M'$ are paired with parts from \mathcal{F}^* (which consist of root edges of M^*). For each part $\pi \in \mathcal{F}^* \subseteq \mathcal{P}^*$ let $E(\pi) \subseteq M''$ denote the M'' -edges that are paired with π . It follows that $\{E(\pi) : \pi \in \mathcal{F}^*\}$ partitions M'' . Using Claim 9 and the definition E_{AS} , we have $\{E_A : (D, A) \in \mathcal{S}\} = \{E(\pi) : \pi \in \mathcal{F}^*\}$, which is a partition of M'' . ■

Components of $E \cap M$ The true-edges of M induce a forest. Consider the connected components in this forest: for each $f \in F$, let $C_f \subseteq V$ denote the vertices connected to f . Note that $\{C_f : f \in F\}$ partitions V . Moreover, the edges in the components C_f partition $E \cap M$.

Components of $E \cap M^*$ Now consider the forest induced by true edges of M^* and direct each edge towards an F^* -vertex (note that each tree in this forest contains exactly one F^* -vertex). Observe that each vertex $v \in V \setminus F^*$ has exactly one out-edge σ_v , and F^* -vertices have none.

For each $f \in F$, define $T_f^* := \{\sigma_v : v \in C_f\}$ the set of out-edges incident from vertices of C_f (see Figure 3). These sets will be used to charge the increases $\text{Tree}(F - D + A) - \text{Tree}(F)$ for different

swaps $(D, A) \in \mathcal{S}$ to (roughly) disjoint parts of M^* . Since $\{C_f : f \in F\}$ partitions V , it follows that $\{T_f^*\}_{f \in F}$ partitions $E \cap M^*$, and:

Claim 12 $\sum_{f \in F} d(T_f^*) = d(E \cap M^*)$.

We are now ready to bound the increase in the Tree cost under each swap of \mathcal{S} .

Lemma 13 For each swap $(D, A) \in \mathcal{S}$,

$$\text{Tree}(F - D + A) - \text{Tree}(F) \leq 2 \cdot \sum_{f \in D} d(T_f^*) - d(E_A).$$

Proof: By Claim 9, $A \subseteq F^*$ is a part in \mathcal{P}^* . Recall that E_A denotes the true-edges of M paired with A ; let F_A denote the root-edges of M paired with A . Then using Theorem 8 it follows that $(M \setminus (E_A \cup F_A)) \cup A$ is a spanning tree in H ; note that A corresponds to a set of root-edges. Hence, the remaining true-edges $S_A := (E \cap M) \setminus E_A$ is a forest with each component containing some center from $(F \setminus F_A) \cup A$. In other words, S_A connects each vertex in V to some vertex of $(F \setminus F_A) \cup A$. For any $f \in F \cup A$ let C'_f denote vertices in the component of S_A containing f . Note that $\{C'_f : f \in (F \setminus F_A) \cup A\}$ is a refinement of the previously defined partition $\{C_f : f \in F\}$ as $S_A \subseteq E \cap M$. Moreover, the edges in the components C'_f partition S_A .

Note that the length of edges in S_A is $\text{Tree}(F) - d(E_A)$: so if each component of S_A already contains some vertex of $F - D + A$, the lemma holds trivially (without the additional T_f^* terms). Otherwise, we will add a set of true edges of cost at most $2 \sum_{f \in D} d(T_f^*)$ so that each resulting component contains some vertex of $F - D + A$: this would complete the proof of the lemma.

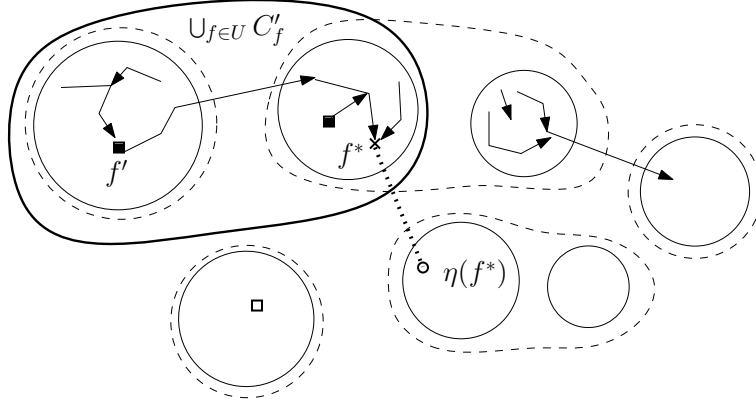
For the proof, it turns out to be convenient to add edges to S_A in two phases. First, we add all the edges in $\bigcup_{f \in D} T_f^*$ to obtain the (true) edge set $S'_A := S_A \cup_{f \in D} T_f^*$. In the second phase, we will add a set N of true edges so that $S'_A \cup N$ connects each D -vertex to some vertex of $F - D + A$. Since S_A already connects all vertices to some vertex in $(F \setminus F_A) \cup A \subseteq F \cup A$, it would follow that each connected component of $S'_A \cup N$ contains some vertex of $F - D + A$, and therefore

$$\text{Tree}(F + A - D) \leq d(S'_A) + d(N) \leq \text{Tree}(F) - d(E_A) + \sum_{f \in D} d(T_f^*) + d(N).$$

To prove the lemma it now suffices to construct a true edge-set N with $d(N) \leq \sum_{f \in D} d(T_f^*)$, such that $S'_A \cup N$ connects each D -vertex to $F - D + A$. The existence of such an edge-set N relies on the metric being the same for both k -median and MST parts of the objective as well as some properties of the set \mathcal{S} of swaps (Claim 14).

Below, we use $\delta(V')$ to denote the edges of S'_A between V' and $V \setminus V'$ for any $V' \subseteq V$. Note that by its definition, each C'_f (for $f \in F \cup A$) is connected in $S_A \subseteq S'_A$.

Constructing N Consider any minimal $U \subseteq D$ such that $\delta\left(\bigcup_{f \in U} C'_f\right) = \emptyset$. If there is no such set U then it follows that each D -vertex is connected in S'_A to some vertex of $(V \setminus \bigcup_{f \in D} C'_f) \cup_{g \in A} C'_g$, and hence to some vertex of $F - D + A$: in this case it suffices to set $N = \emptyset$. Now, suppose there is such a minimal set U . By minimality and the fact that each C'_f is connected in S'_A , it follows that $\bigcup_{f \in U} C'_f$ is connected in S'_A . We need the following claim (see also an example in Figure 4).



Squares denote vertices in D ; here $|D| = 3$. Solid squares are vertices in $U \subseteq D$; here $|U| = 2$. Components $\{C_f : f \in F\}$ are the dashed regions. Solid lines are edges in $\cup_{f \in U} T_f$. Components $\{C'_f : f \in F \cup A\}$ are the solid circles. Dotted edge $(f^*, \eta(f^*))$ is added to set N .

Figure 4: Addition of edges to set N .

Claim 14 *There exists $f^* \in F^* \cap \left(\bigcup_{f \in U} C'_f\right)$ and $f' \in U$ such that $\bigcup_{f \in U} T_f^*$ contains a path between f' and f^* .*

Proof: Let any $f' \in U$. Consider the directed path P from f' obtained by following *out-edges* σ until the *first occurrence* of a vertex v that is either in F^* or in $V \setminus \left(\bigcup_{f \in U} C'_f\right)$. Since F^* -vertices are the only ones with no out-edge σ , and the set of all out-edges $\{\sigma_w : w \in V\} = E \cap M^*$ is acyclic, there must exist such a vertex $v \in F^* \cup \left(V \setminus \left(\bigcup_{f \in U} C'_f\right)\right)$.

We first claim that $P \subseteq \bigcup_{f \in U} T_f^*$. To see this, observe that $C'_f \subseteq C_f$ for all $f \in D \supseteq U$: recall that C_s (resp. C'_s) are the connected components in $E \cap M$ (resp. $S_A \subseteq E \cap M$). So $P \subseteq \{\sigma_w : w \in \bigcup_{f \in U} C'_f\} \subseteq \{\sigma_w : w \in \bigcup_{f \in U} C_f\} = \bigcup_{f \in U} T_f^*$.

Suppose now (for a contradiction) that vertex $v \notin F^*$. Then $v \in V \setminus \left(\bigcup_{f \in U} C'_f\right)$, but this implies $\delta\left(\bigcup_{f \in U} C'_f\right) \neq \emptyset$ since path $P \subseteq \bigcup_{f \in U} T_f^* \subseteq S'_A$ leaves $\bigcup_{f \in U} C'_f$. So $v \in F^* \cap \left(\bigcup_{f \in U} C'_f\right)$ and $P \subseteq \bigcup_{f \in U} T_f^*$ is a path from f' to v . ■

Consider f^* and f' as given in Claim 14. If $f^* \in A$ then the component $\bigcup_{f \in U} C'_f$ of S'_A is already connected to $F - D + A$. Otherwise, by Claim 7 (using properties of the swaps \mathcal{S}) we have $\eta(f^*) \in F \setminus D$, and we add edge $(f^*, \eta(f^*))$ to N which connects component $\bigcup_{f \in U} C'_f$ to $\eta(f^*) \in F - D \subseteq F - D + A$. Now, $d(f^*, \eta(f^*)) \leq d(f^*, f') \leq \sum_{f \in U} d(T_f^*)$; the last inequality uses Claim 14 and the fact that the metric d is the same for both k -median and MST parts of the objectives. This is the only point in the proof where we use uniformity in the metrics for the k -median and MST objectives.² In either case, U gets connected to $F - D + A$ in $S'_A \cup N$, and the cost of N increases by at most $\sum_{f \in U} d(T_f^*)$.

²As the example in Section B shows, having the same metric is crucial for a constant locality gap.

We apply the above argument to every minimal $U \subseteq D$ with $\delta\left(\bigcup_{f \in U} C'_f\right) = \emptyset$. The increase in cost of N due to each such U is at most $\sum_{f \in U} d(T_f^*)$. Since such minimal sets U are *disjoint*, we have $d(N) \leq \sum_{f \in D} d(T_f^*)$. Clearly $S'_A \cup N$ connects each D -vertex to $F - D + A$. ■

Using Lemma 13 for each $(D, A) \in \mathcal{S}$ weighted by the coefficient $\alpha(D, A)$ from Theorem 5 and adding them together, we obtain:

$$\begin{aligned} & \sum_{(D,A) \in \mathcal{S}} \alpha(D, A) \cdot [\text{Tree}(F - D + A) - \text{Tree}(F)] \\ \leq & 2 \cdot \sum_{(D,A) \in \mathcal{S}} \alpha(D, A) \cdot \sum_{f \in D} d(T_f^*) - \sum_{(D,A) \in \mathcal{S}} \alpha(D, A) \cdot d(E_A) \end{aligned} \quad (4)$$

$$= 2 \sum_{f \in F} \left(\sum_{(D,A) \in \mathcal{S}: f \in D} \alpha(D, A) \right) \cdot d(T_f^*) - \sum_{e \in M''} \left(\sum_{(D,A) \in \mathcal{S}: e \in E_A} \alpha(D, A) \right) \cdot d_e \quad (5)$$

$$\leq 2 \left(1 + \frac{1}{t}\right) \sum_{f \in F} d(T_f^*) - \sum_{e \in M''} d_e \quad (6)$$

$$= 2 \left(1 + \frac{1}{t}\right) \cdot d(E \cap M^*) - d(M'') \quad (7)$$

Above (4) is by Lemma 13, (5) is by interchanging summations using the fact that $E_A \subseteq M''$ (for all $(D, A) \in \mathcal{S}$) from Claim 11. The first term in (6) uses the property in Theorem 5 that each $f \in F$ is dropped (i.e. $f \in D$) to a total extent at most $1 + \frac{1}{t}$; the second term uses

$$\sum_{(D,A) \in \mathcal{S}: e \in E_A} \alpha(D, A) = 1 \quad \text{for each } e \in M'',$$

which follows by Claim 11 and the property in Theorem 5 that each $f^* \in F^*$ is added to a total extent of one in \mathcal{S} . Finally (7) is by Claim 12.

Adding the inequality $0 \leq d(E \cap M^*) - d(M')$ from Claim 10 yields:

$$\sum_{(D,A) \in \mathcal{S}} \alpha(D, A) \cdot [\text{Tree}(F - D + A) - \text{Tree}(F)] \leq \left(3 + \frac{2}{t}\right) \cdot d(E \cap M^*) - d(E \cap M),$$

since M' and M'' partition the true edges $E \cap M$. Thus we obtain Inequality (2).

4 Conclusion

In this paper we obtained the first constant-factor approximation algorithm for k -location capacitated vehicle routing, a basic location-routing problem. Our algorithm used existing lower bounds for capacitated vehicle routing to reduce to a new location problem, k median forest, which generalizes the k -median and k -tree problems. Finally we obtained a local-search based $(3 + \epsilon)$ -approximation algorithm for k median forest. Both the k -location capacitated vehicle routing and k median forest problems generalize k -median: so they are hard to approximate within a factor of $1 + 2/e \approx 1.736$ [14]. Improving either the approximation ratio or the hardness result would be interesting. We mention here two potential avenues for improving on our approximation ratio:

- Given that both the k -median and k -tree problems admit approximation ratios better than 3, is there a better than 3 approximation ratio for k median forest as well? A possible first step in this direction is to combine the “Lagrangian multiplier preserving” algorithms for k -median [14] and k -tree [7], which are currently based on different methods.
- The explicit reduction to k median forest loses a factor of 4 in the approximation ratio. Can we obtain an algorithm that avoids such a reduction? This is likely to require a better lower bound for the location-routing problem.

Acknowledgement: We thank the anonymous reviewers for their comments which helped improve the presentation and discussion in the paper.

References

- [1] K. Altinkemer and B. Gavish. Heuristics for unequal weight delivery problems with a fixed error guarantee. *Operations Research Letters*, 6:149–158, 1987.
- [2] K. Altinkemer and B. Gavish. Heuristics for delivery problems with constant error guarantees. *Transportation Research*, 24:294–297, 1990.
- [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 J. Comput.*, 33(3):544–562, 2004.
- [4] A. Balakrishnan, J. Ward, and R. Wong. Integrated facility location and vehicle routing models: Recent work and future prospects. *Amer. J. Mathematical and Management Sciences*, 7:35–61, 1987.
- [5] O. Berman, P. Jaillet, and D. Simchi-Levi. Location-routing problems with uncertainty. In Z. Drezner, editor, *Facility Location: A Survey of Applications and Methods*, pages 427–452. Springer, 1995.
- [6] J. Byrka, T. Pensyl, B. Rybicki, A. Srinivasan, and K. Trinh. An improved approximation for k -median, and positive correlation in budgeted optimization. *CoRR*, abs/1406.2951, 2014.
- [7] T. Carnes and D. B. Shmoys. Primal-dual schema and lagrangian relaxation for the k -location-routing problem. In *APPROX-RANDOM*, pages 99–110, 2011.
- [8] M. Charikar, S. Guha, E. Tardos, and D. B. Shmoys. A constant-factor approximation algorithm for the k -median problem. *J. Comput. Syst. Sci.*, 65(1):129–149, 2002.
- [9] M. X. Goemans and D. P. Williamson. A general approximation technique for constrained forest problems. *SIAM J. Comput.*, 24(2):296–317, 1995.
- [10] I. L. Gørtz and V. Nagarajan. Locating depots for capacitated vehicle routing. *CoRR*, abs/1103.0985, 2011.
- [11] A. Gupta and K. Tangwongsan. Simpler analyses of local search algorithms for facility location. *CoRR*, abs/0809.2554, 2008.

- [12] M. Haimovich and A. H. G. R. Kan. Bounds and heuristics for capacitated routing problems. *Mathematics of Operations Research*, 10(4):527–542, 1985.
- [13] T. Harks, F. G. König, and J. Matuschke. Approximation algorithms for capacitated location routing. *Transportation Science*, 47(1):3–22, 2013.
- [14] K. Jain, M. Mahdian, and A. Saberi. A new greedy approach for facility location problems. In *STOC*, pages 731–740, 2002.
- [15] K. Jain and V. V. Vazirani. Approximation algorithms for metric facility location and k -median problems using the primal-dual schema and lagrangian relaxation. *J. ACM*, 48(2):274–296, 2001.
- [16] R. Krishnaswamy, A. Kumar, V. Nagarajan, Y. Sabharwal, and B. Saha. The matroid median problem. In *SODA*, 2011.
- [17] G. Laporte. Location-routing problems. In B. Golden and A. Assad, editors, *Vehicle Routing: Methods and Studies*, pages 163–198. North-Holland, 1988.
- [18] G. Laporte. A survey of algorithms for location-routing problems. *Investigación Operativa*, 1:93–123, 1989.
- [19] C. Li and D. Simchi-Levi. Worst-case analysis of heuristics for multidepot capacitated vehicle routing problems. *ORSA Journal on Computing*, 2(1):64–73, 1990.
- [20] S. Li and O. Svensson. Approximating k -median via pseudo-approximation. In *STOC*, pages 901–910, 2013.
- [21] H. Min, V. Jayaraman, and R. Srivastava. Combined location-routing problems: A synthesis and future research directions. *Eur. J. Oper. Res.*, 108:1–15, 1998.
- [22] G. Nagy and S. Salhi. Location-routing: Issues, models and methods. *Eur. J. Oper. Res.*, 177(2):649–672, 2007.
- [23] A. Schrijver. *Combinatorial Optimization*. Springer, 2003.
- [24] P. Toth and D. Vigo, editors. *The vehicle routing problem*. SIAM Monographs on Discrete Mathematics and Applications, Philadelphia, PA, USA, 2002.
- [25] V. V. Vazirani. *Approximation algorithms*. Springer, 2001.
- [26] D. P. Williamson and D. B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, 2011.

A Example Comparing k -Median, k -Tree and k Median Forest

Here, we give an example which shows that near-optimal solutions to the k -median, k -tree and k median forest problems can be very far from each other. This implies that any approximation algorithm for k median forest must simultaneously take into account both the median and tree parts of its objective.



Figure 5: Instance for which the values of near-optimal solutions to k -median, k -tree and k median forest problems are far apart for $k = 4$. We have vertex weights $q(u_1) = q(u_2) = q(v_1) = q(v_2) = \ell^4$ and $q(u_0) = q(v_0) = 1$.

The underlying metric consists of six vertices $\{u_0, u_1, u_2\} \cup \{v_0, v_1, v_2\}$, see Figure 5. Let ℓ be a parameter that will be set to be arbitrarily large. The distance between any u_i and v_j (for all $i, j \in \{0, 1, 2\}$) is infinite; $d(u_0, u_1) = d(u_0, u_2) = \ell^3$, $d(u_1, u_2) = \ell^2$; and $d(v_0, v_1) = d(v_0, v_2) = \ell^4$, $d(v_1, v_2) = \ell$. The weights of vertices are $q(u_1) = q(u_2) = q(v_1) = q(v_2) = \ell^4$ and $q(u_0) = q(v_0) = 1$. The bound $k = 4$ and parameter $\rho = \ell^2$ for the k median forest problem. Let \mathcal{S}_{med} , \mathcal{S}_{tree} and \mathcal{S}_{kmf} denote the sets of solutions that have objective values within an $o(\ell)$ factor of the respective optimum for the k -median, k -tree and k median forest objectives. We claim that \mathcal{S}_{med} , \mathcal{S}_{tree} and \mathcal{S}_{kmf} are mutually disjoint.

It can be checked directly that the optimal k -median value is $\ell^3 + \ell^4 \leq 2\ell^4$. Moreover the only solution of value $o(\ell^5)$ is $\{u_1, u_2, v_1, v_2\}$; so \mathcal{S}_{med} consists of just this solution.

The optimal k -tree value is $\ell + \ell^2 \leq 2\ell^2$. For any solution $F \in \mathcal{S}_{tree}$ (i.e. having value $o(\ell^3)$), we must have $u_0, v_0 \in F$, $|F \cap \{u_1, u_2\}| = 1$ and $|F \cap \{v_1, v_2\}| = 1$. So \mathcal{S}_{tree} consists of the 4 solutions: $\{u_0, v_0, u_i, v_j\}$ where $i, j \in \{1, 2\}$.

For the k median forest objective, it can be seen that the optimal value is $\rho \cdot (\ell^3 + \ell) + \ell^4 \cdot \ell + \ell^3 = 2(\ell^5 + \ell^3)$; from the solutions $\{u_1, u_2, v_0, v_1\}$ and $\{u_1, u_2, v_0, v_2\}$. Moreover, any other solution has value $\Omega(\ell^6)$; so \mathcal{S}_{kmf} consists of the above two solutions.

Clearly \mathcal{S}_{med} , \mathcal{S}_{tree} and \mathcal{S}_{kmf} are disjoint.

B Large Locality Gap for Non-Uniform k Median Forest

Here we consider the k median forest problem with two *distinct* metrics for the k -median and MST parts. In this problem, there is a set of vertices V with weights $\{q_u\}_{u \in V}$, two metrics d and c defined on V , and a bound k . The goal is to find $S \subseteq V$ with $|S| = k$ minimizing

$$\sum_{u \in V} q_u \cdot d(u, S) + c(\text{MST}(V/S)).$$

Here $c(\text{MST}(V/S))$ is the length (under metric c) of the minimum spanning tree in the graph obtained by contracting S to a single vertex. We give an example below showing an unbounded locality gap for this problem, which is in sharp contrast to the k median forest problem under a single metric.

Lemma 15 *The locality gap of non-uniform k median forest with multi-swaps is unbounded.*

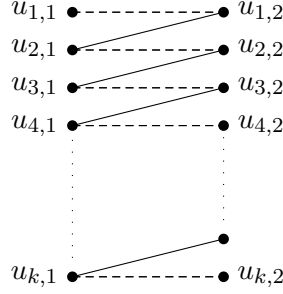


Figure 6: Metric c : Dashed edges have weight 0, all other edges have weight M . Metric d : The solid edges have weight 0, all other edges have weight 1.

Proof: Fix values $M \gg w \gg 1$. Let $V = \{u_{i,j} : i \in [k], j \in \{1, 2\}\}$, so $|V| = 2k$. Define vertex-weights as follows: $q(u_{k,2}) = 1$ and all other vertices have weight w . The metric d for the k -median part is:

$$d(x, y) = \begin{cases} 0 & \text{if either } x = y \text{ or } \{x, y\} = \{u_{i,2}, u_{i+1,1}\} \text{ for some } i \in [k-1] \\ 1 & \text{otherwise} \end{cases}$$

The metric c for the MST part of the objective is:

$$c(x, y) = \begin{cases} 0 & \text{if either } x = y \text{ or } \{x, y\} = \{u_{i,1}, u_{i,2}\} \text{ for some } i \in [k] \\ M & \text{otherwise} \end{cases}$$

See Figure 6 for an example. Observe that for any $S \subseteq V$ with $|S| = k$, we have $c(\text{MST}(V/S)) < M$ if and only if $|S \cap \{u_{i,1}, u_{i,2}\}| = 1$ for all $i \in [k]$. So the non-uniform k median forest objective is smaller than M only if $|S \cap \{u_{i,1}, u_{i,2}\}| = 1, \forall i \in [k]$.

We claim that the optimal value is at most one. Consider the solution $S^* = \{u_{i,1}\}_{i=1}^k$. It is clear that $c(\text{MST}(V/S^*)) = 0$. Moreover, $\sum_{u \in V} q(u) \cdot d(u, S^*) = 1$ with vertex $u_{k,2}$ being the only contributor.

We now claim that the solution $L = \{u_{i,2}\}_{i=1}^k$ is locally optimal under even $(k-1)$ -swaps. First, observe that $c(\text{MST}(V/L)) = 0$ and $\sum_{u \in V} q(u) \cdot d(u, L) = w$ with vertex $u_{1,1}$ being the only contributor. So L has objective value of w . Secondly, notice that every solution S obtained by some $(k-1)$ -swap of L has either MST-objective of M or median-objective of w . Thus L is a local optimum and the locality gap is $w \gg 1$. ■

This example is similar to the locality gap shown in [16] for the *matroid median* problem. It turns out that one can also obtain a constant-factor approximation algorithm for non-uniform k median forest, along the lines of [16]; see [10] for details.