Lagrangian relaxation for the k-median problem: new insights and continuity properties

Aaron Archer[⋆], Ranjithkumar Rajagopalan[⋆⋆], and David B. Shmoys[⋆⋆⋆]

School of Operations Research and Industrial Engineering
Cornell University, Ithaca, NY 14853
{aarcher,rajagop}@orie.cornell.edu, shmoys@cs.cornell.edu

Abstract. This work gives new insight into two well-known approximation algorithms for the uncapacitated facility location problem: the primal-dual algorithm of Jain & Vazirani, and an algorithm of Mettu & Plaxton. Our main result answers positively a question posed by Jain & Vazirani of whether their algorithm can be modified to attain a desired "continuity" property. This yields an upper bound of 3 on the integrality gap of the natural LP relaxation of the k-median problem, but our approach does not yield a polynomial time algorithm with this guarantee. We also give a new simple proof of the performance guarantee of the Mettu-Plaxton algorithm using LP duality, which suggests a minor modification of the algorithm that makes it Lagrangian-multiplier preserving.

1 Introduction

Facility location problems have been widely studied in both the operations research and computer science literature We consider the two most popular variants of facility location: the k-median problem and the uncapacitated facility location problem (UFL). In both cases, we are given a set $\mathcal{C}$ of *clients* who must be served by a set $\mathcal{F}$ of *facilities*, and distances c_{ij} for all $i, j \in \mathcal{F} \cup \mathcal{C}$. When $i \in \mathcal{F}$ and $j \in \mathcal{C}$, c_{ij} is the cost of serving client j from facility i. We assume that these distances form a semi-metric; that is, $c_{ij} = c_{ji}$, and $c_{ik} \leq c_{ij} + c_{jk}$ for all $i, j, k \in \mathcal{F} \cup \mathcal{C}$. The goal is to open some subset of facilities $S \subseteq \mathcal{F}$ in order to minimize the total *connection cost* of serving each client from its closest facility, subject to some limitations on S. Whereas k-median imposes the hard constraint $|S| \leq k$, in UFL we have *facility costs* f_i for all $i \in \mathcal{F}$, and we aim to minimize the sum of the facility and connection costs.

Both problems are NP-hard, so we are interested in obtaining approximation algorithms. An α-*approximate* solution is one whose objective function is within a factor of α of the optimal solution. An α-*approximation* algorithm is one that runs in polynomial time and always returns an α-approximate solution. One primary theme of this line of research is to exploit a classical linear programming (LP) relaxation of the problem, initially proposed by Balinski [4]. We contribute to this vein by shedding new light on

[⋆] Supported by the Fannie and John Hertz Foundation.
[⋆⋆] Research partially supported by NSF grant CCR-9912422.
[⋆⋆⋆] Research partially supported by NSF grant CCR-9912422.

two existing UFL algorithms, the primal-dual algorithm of Jain & Vazirani (JV) [12], and the algorithm of Mettu & Plaxton (MP) [19].

We show that the JV algorithm can be made "continuous," resolving a question posed in [12]. Because of their results connecting the k-median and UFL problems via Lagrangian relaxation, our result proves that the integrality gap of the most natural LP relaxation for k-median is at most 3, improving the previous best upper bound of 4 [7]. Since our algorithm involves solving the NP-hard maximum independent set problem, it does not lead directly to a polynomial-time 3-approximation algorithm; nonetheless, we believe that it is a significant step in that direction.

Mettu & Plaxton [19] prove that their algorithm achieves an approximation factor of 3, but their analysis never explicitly mentions an LP. Because the MP and JV algorithms appear superficially to be very similar and both achieve a factor of 3, many researchers wondered whether there was a deeper connection. We exhibit a dual solution that proves the MP primal solution is within a factor of 3 of the LP optimum. Interpreting their algorithm within an LP framework yields an additional benefit: it highlights that a slight modification of MP also satisfies the *Lagrangian-multiplier preserving* (LMP) property, which was not previously known. We note that Pál & Tardos independently constructed the same dual solution for use in creating cross-monotonic cost-sharing methods for facility location in a game-theoretic context [20].

The UFL problem has been studied from many perspectives since the 1960's, but the first approximation algorithm was given much later by Hochbaum [11], who achieved an $O(\log |\mathcal{C}|)$ factor using a method based on greedy set cover. Shmoys, Tardos & Aardal [21], gave the first constant factor of 3.16. A series of papers have improved this to 1.52, by Mahdian, Ye & Zhang [16]. In the process, many and varied techniques have been brought to bear on the problem, and the insights gained have been applied elsewhere. Most prominent among the algorithmic and analytical techniques used have been LP rounding, filtering, various greedy algorithms, local search, primal-dual methods, cost-scaling, and dual fitting [7, 9, 10, 17, 12, 14, 21, 22]. Guha & Khuller [10] showed that UFL cannot be approximated to a factor better than 1.463 unless $P = NP$.

K-median seems to be more difficult. The best hardness bound known is $1 + \frac{2}{e}$ [17], and the standard LP relaxation has an integrality gap of at least 2. Lin & Vitter [15] gave a constant-factor bicriterion approximation algorithm, and Bartal [6, 5] achieved a near-logarithmic factor via probabilistic tree-embeddings, but the first constant factor of $6\frac{2}{3}$ was given by Charikar, Guha, Tardos & Shmoys [8], who used LP rounding. This factor was improved to 6 by Jain & Vazirani [12], 4 by Charikar & Guha [7], and $(3 + \epsilon)$ by Arya et al. [3]. The factor of 4 is attained via a refinement of the work of Jain & Vazirani, while the $(3 + \epsilon)$ is completely different, using local search.

Basic economic reasoning shows a connection between UFL and k-median. Consider a uniform facility cost z in the UFL. When $z = 0$, the best solution opens all facilities. As z increases from zero, the number of open facilities in the optimal solution decreases monotonically to one. Suppose some value of z causes the optimal UFL solution to open exactly k facilities S. Then S is also the optimal k-median solution.

Jain & Vazirani [12] exploit this relationship by interpreting the standard LP relaxation of UFL as the Lagrangian relaxation of the LP for k-median. Their elegant primal-dual UFL algorithm achieves a guarantee of 3, and also satisfies the LMP property. They then show how to convert any LMP algorithm into an approximation algorithm for k-median while losing an additional factor of 2 in the guarantee. More importantly for us, they show that the solution S output by their UFL algorithm is a 3-approximate solution for the $|S|$-median problem. Thus, if one can find, in polynomial time, a value of z such that the JV algorithm opens exactly k facilities, this constitutes a 3-approximation algorithm for the k-median problem. Sadly, there are inputs for which no value of z causes the JV algorithm (as originally stated) to open exactly k facilities.

We modify the JV algorithm to attain the following *continuity* property. Consider the solution $S(z)$ output by the algorithm, as a function of the uniform facility cost z. As z changes, we ensure that $|S(z)|$ never jumps by more than 1. Since the algorithm opens all facilities when $z = 0$ and only one when z is sufficiently large, there is some value for which it opens exactly k. By standard methods (either binary search or Megiddo's parametric search [18]), we can find the desired value using a polynomial number of calls with different values of z. This appears to answer the question posed in [12]. Unfortunately, our algorithm involves finding maximum independent sets, which is NP-hard. This leaves the open question of whether one can achieve a version of JV that has the continuity property *and* runs in polynomial time.

There are two rays of light. First, our algorithm does prove that the integrality gap of the standard k-median LP is at most 3. This was not known before, and it is novel because most proofs that place upper bounds on the integrality gaps of LP relaxations rely on polynomial-time algorithms. (For an interesting exception to this rule, see [2].) Second, it is enough to compute maximal independent sets that are continuous with respect to certain perturbations of the graph [1]. The only types of sets that we know to be continuous are maximum independent sets, but we are hopeful that one could compute, in polynomial time, some other type of continuous maximal independent set.

2 Ensuring continuity in Jain-Vazirani

The JV algorithm for UFL is based on the following standard LP relaxation for the problem (originally proposed by Balinski [4]), and its dual.

Primal LP:

$$\begin{array}{ll} \text{min} & \sum_{i\in\mathcal{F}} f_i y_i + \sum_{ij\in\mathcal{F}\times\mathcal{C}} c_{ij}x_{ij} \\ \text{such that:} & \sum_{i\in\mathcal{F}} x_{ij} = 1 \quad \forall j \in \mathcal{C} \\ & y_i - x_{ij} \geq 0 \; \forall ij \in \mathcal{F}\times\mathcal{C} \\ & y_i, x_{ij} \geq 0 \quad \forall ij \in \mathcal{F}\times\mathcal{C} \end{array}$$

Dual LP:

$$\begin{array}{ll} \text{max} & \sum_{j\in\mathcal{C}} v_j \\ \text{such that:} & \sum_{j\in\mathcal{C}} w_{ij} \leq f_i \quad \forall i \in \mathcal{F} \\ & v_j - w_{ij} \leq c_{ij} \; \forall ij \in \mathcal{F}\times\mathcal{C} \\ & v_j, w_{ij} \geq 0 \quad \forall ij \in \mathcal{F}\times\mathcal{C} \end{array}$$

Adding constraints $y_i, x_{ij} \in \{0,1\}$ gives an exact IP formulation. The variable y_i indicates whether facility i is open, and x_{ij} says whether client j is connected to facility i. Intuitively, v_j is the total amount of money that client j is willing to pay to be served: w_{ij} is its share towards the cost of facility i, and the rest pays for its connection cost.

[1] The class of perturbations needs to be defined carefully; an earlier version of this abstract proposed one that was more general than necessary, and implied that the only possible realization of this approach required a maximum independent set.

The JV algorithm operates in two phases. Phase I consists of growing the dual variables, maintaining dual feasibility, and gradually building a primal solution until that solution is feasible. Phase II is a cleanup phase in which we keep only a subset of the facilities opened in phase I. This results in the following theorem.

Theorem 1 (Jain-Vazirani 2001) *The Jain-Vazirani facility location algorithm yields a feasible integer primal solution and a feasible dual solution to the UFL LP, satisfying $C + 3F \leq 3\sum_{j \in C} v_j \leq 3OPT$, where OPT is the value of the optimal UFL solution.*

We now describe the algorithm precisely but conceptually, motivating each step but ignoring the implementation details. We envision dual and primal solutions changing over time. At time zero, we set all primal and dual variables to zero, so the dual is feasible and the primal is infeasible. Throughout phase I, we maintain dual feasibility and work towards primal feasibility. We also enforce primal complementary slackness, meaning that we never open a facility i unless it is fully paid for by the dual variables (i.e., $\sum_j w_{ij} = f_i$) and we connect client j to facility i only if $v_j = c_{ij} + w_{ij}$, i.e., j's dual variable fully pays for connection cost and its share of facility i's cost.

We initially designate all clients as *active*, and raise their dual variables at unit rate. Eventually, some edge ij goes *tight*, meaning that $v_j = c_{ij}$, i.e., client j's dual variable has completely paid for its connection cost to facility i. We continue raising the v_j variables at unit rate for all active clients j, but now we must also raise the w_{ij} cost shares for all tight edges ij. Eventually, we pay for and open some facility i when the constraint $\sum_j w_{ij} \leq f_i$ goes tight. Now we must freeze all of the cost shares w_{ij} in order to maintain dual feasibility, so we must also freeze the dual variable v_j for every client j with a tight edge to facility i. Fortunately, facility i is now open, so we can assign client j to be served by facility i and declare it *inactive*. We refer to facility i as client j's *connecting witness*. Conveniently, v_j exactly pays for j's connection cost plus its share of facility i's cost, since $v_j = c_{ij} + w_{ij}$. We continue in this manner. It can also occur that an active client gains a tight edge to a facility that is already open. In this case, the client is immediately connected to that facility. Phase I terminates when the last active client is connected. If any combination of events is set to occur simultaneously, we can break ties in an arbitrary order. Notice that the tiebreaking rule has no effect on the dual solution generated.

At the end of phase I, we have some preliminary set S_0 of open facilities. As we have mentioned, the algorithm opens a facility only when the w_{ij} variables fully pay for it, and the v_j variable for client j exactly pays for its connection cost plus its share of the facility cost for its connecting witness. Then why is S_0 not an optimal solution? It is because some client may have contributed a non-zero cost share to some open facility to which it is not connected. Thus, we must clean up the solution to avoid this problem. In phase II, we select a subset of facilities $S \subseteq S_0$ so that each client pays a positive cost share to at most one open facility. Every client that has a tight edge to a facility in S is said to be *directly connected*. Thus, the directly connected clients exactly pay for their own connection costs and all of the facility costs. The trick is, each client j that is not directly connected must still be connected to some facility i. We obtain a 3-approximation algorithm if we can guarantee that $c_{ij} \leq 3v_j$.

Phase II proceeds as follows. We construct a graph G with vertices S_0, and include an edge between $i, k \in S_0$ if there exists some client j such that $w_{ij}, w_{kj} > 0$. We must select S to be an independent set in G. Otherwise, some client j offered cost shares to two facilities in S, but it can afford to pay for only one. We might as well choose S to be a *maximal* independent set (meaning that no superset of S is also an independent set, so every vertex in $S_0 - S$ is adjacent to a vertex in S). For each client j that is not directly connected, consider its connecting witness i. Since $i \notin S$, there must exist an adjacent facility $k \in S$, so we connect j to k. This completes the description of the algorithm.

In their original paper [13], Jain & Vazirani chose a particular set S, but in the journal version, they modify their analysis to accommodate any maximal independent set. Later, we will choose a maxi*mum* (cardinality) independent set, but for Theorem 1 and the present discussion, any maximal independent set suffices.

The LMP property becomes important when we view the LP relaxation of UFL as the Lagrangian relaxation of the standard LP relaxation of k-median. The k-median LP is the same as the UFL LP, except there is no facility cost term in the objective, and we add the constraint $\sum_i y_i \leq k$. By Lagrangian relaxation, we mean to remove the cardinality constraint, set a non-negative penalty parameter z, and add the term $z(\sum_i y_i - k)$ to the objective function. This penalizes solutions that violate the constraint by opening more than k facilities, and gives a bonus to solutions that open fewer than k. Aside from the constant term of $-zk$, this is precisely the same as the LP relaxation of UFL, setting all facility costs to z. Notice that the objective function matches the true k-median objective whenever exactly k facilities are opened. Thus, every feasible solution for the original k-median LP is also feasible for its Lagrangian relaxation, and the objective function value in the relaxation is no greater than in the original LP. Therefore, every dual feasible solution for the Lagrangian relaxation provides a lower bound on the optimal k-median solution. These observations lead to the following result in [12].

Theorem 2 *Suppose that we set all facility costs to $z > 0$, so that the JV algorithm opens exactly k facilities. Then this is a 3-approximate k-median solution.*

A bad example and how to fix it in general. We first give a well-known example showing that the JV algorithm as described above does not satisfy the continuity property. We then show that perturbing the input fixes this bad example. Our main result shows that this trick works in general. Consider the metric space given by a star with h arms, each of length 1. At the end of each arm there is one client j and one potential facility i_j. There is also one facility (called i_0) located at the hub of the star. (See Figure 1, setting all $\epsilon_j = 0$.) When $z < 1 + \frac{1}{h-1}$, each client completely pays for the facility located on top of it by time z, while the hub facility has still not been paid for. Hence, $G(z)$ consists of these h facilities, with no edges between them. When $z > 1 + \frac{1}{h-1}$, the hub is opened and all clients connected to it before time z, so $G(z)$ has just one vertex, the hub. Thus, $|S(z)|$ jumps from h down to 1 at the critical value $z = 1 + \frac{1}{h-1}$.

Now perturb the instance by an arbitrarily small amount, moving each client j out past its nearby facility by an amount $\epsilon_j \ll 1$, where $0 = \epsilon_1 < \ldots < \epsilon_h$. Let $\epsilon = \sum_{j=1}^{h} \epsilon_j$, and let $z_1 = \frac{h+\epsilon}{h-1}$. For $z > z_1$, the hub facility is opened before any of the arm facilities, so $G(z)$ is just one isolated vertex. At the critical value $z = z_1$, the hub facility is paid for at exactly the same moment as facility 1. For slightly smaller values

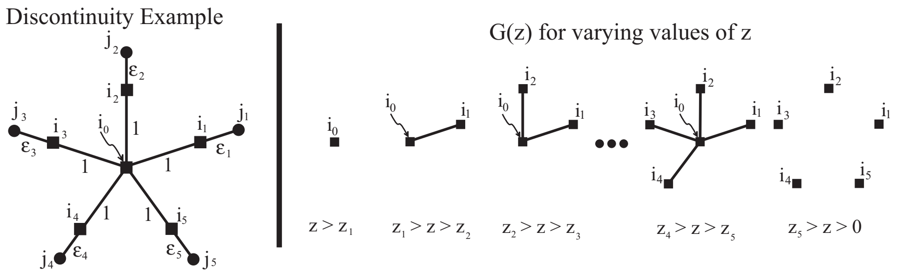

Fig. 1. Discontinuity example (with $h = 5$) and its perturbation.

of z, facility 1 is paid for first, then the hub is opened before any other facility is paid for. Clearly, there exist some $z_1 > z_2 > \ldots > z_h > 0$ such that when $z \in (z_{i+1}, z_i)$, facilities $1, \ldots, i$ are opened before the hub in phase I, and facilities $(i+1), \ldots, h$ are not opened. For z in this range, $G(z)$ consists of the hub facility with edges to facilities $1, \ldots, i$, because client j contributes toward the costs of both the hub and the open facility j, for $1 \leq j \leq i$. For $z \in [0, z_h)$, $G(z)$ contains just isolated vertices $i_1, \ldots, i_h$.

Theorem 1 holds no matter which maximal independent set we choose in phase II, so let $S(z)$ be a maximum independent set. When $z \in (z_{i+1}, z_i)$, $S(z)$ consists of the i facilities $1, \ldots, i$. Thus, $|S(z)|$ changes by at most one at each of the critical values. We have made JV algorithm continuous by perturbing the input an arbitrarily small amount. Our main result is that this trick always works.

We now give some definitions to make our claim precise. We also state our two main results, Theorems 3 and 4, but prove them later. An *event* of the algorithm is the occurrence that either an edge ij goes tight (because client j is active at time c_{ij}) or some facility becomes paid for in phase I. We say that an instance of the UFL problem is *degenerate* if there is some time at which three or more events coincide, or there are at least two points in time where two events coincide. An instance of the k-median problem is degenerate if there exists some $z > 0$ that yields a degenerate UFL instance. (For every non-trivial instance, it is easy to select z so that there is *one* time when two events coincide.) Notice that an instance of the k-median problem simply consists of the distances $\{c_{ij} : ij \in \mathcal{F} \times \mathcal{C}\}$, so we consider an instance to be a point in $\mathbb{R}_+^{\mathcal{F} \times \mathcal{C}}$.

Theorem 3 *The set of all degenerate instances to the k-median problem has Lebesque measure zero.*

For a non-degenerate UFL instance, let us define the *trace* of the algorithm to be the sequence of events encountered during phase I. Notice that $G(z)$ (and consequently, $|S(z)|$) depends only on the trace. Define z_0 to be a *critical* value of z if, when $z = z_0$, there is some point in time where at least two events coincide. For a graph G, let $\mathcal{I}(G)$ denote the size of the largest independent set in G.

Theorem 4 *As z passes through a critical value at which only two events coincide, $\mathcal{I}(G(z))$ changes by at most* 1.

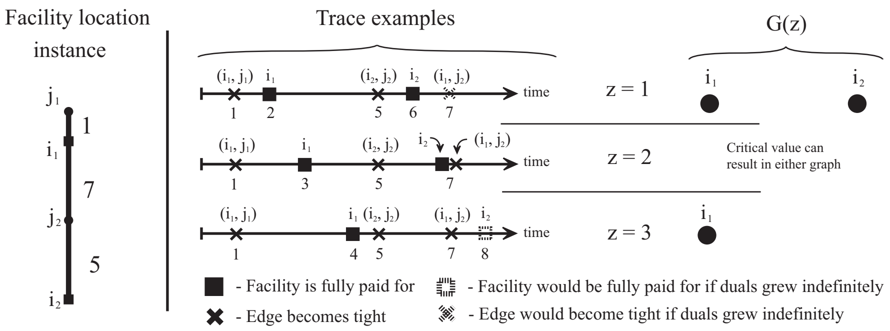

Fig. 2. Trace example.

As we will show, this holds because $G(z)$ changes only slightly when z passes through a non-degenerate critical value. Thus, the algorithm is continuous if our k-median instance is non-degenerate.

Example of traces. To clarify the concept of a trace, we give three traces for the simple facility location instance in Figure 2. When $z = 1$, both i_1 and i_2 are opened, j_1 is connected to i_1 and j_2 is connected to i_2. The edge (i_1, j_2) would become tight at time 7 if v_{j_2} were allowed to grow indefinitely, but v_{j_2} stops growing at time 6, when j_2 is connected to i_2. When $z = 3$, j_1 pays to open i_1 and j_2 connects to it before i_2 is paid for. The figure shows that i_2 would have opened at time 8 if v_{j_3} were allowed to continue growing. At the critical value $z = 2$, i_2 is paid for at the same time that (i_1, j_2) becomes tight, so tiebreaking determines which of the previous solutions is output. The final output of the algorithm depends only on the order of events, not on the actual times. Thus, as z changes, events may slide forward and backward on the trace, but the output changes only at critical values, when events change places.

Exploiting non-degeneracy. For a non-degenerate instance of k-median, we wish to understand how $G(z)$ changes when z passes through a critical value, as summarized in the following theorem.

Theorem 5 *When z passes through a critical value where exactly two events coincide, the graph $G(z)$ can change only in one of the following ways: (a) a single existing facility is deleted (along with its incident edges), (b) a single new facility is added, along with edges to one or more cliques of existing facilities, (c) a single existing facility gains edges to one clique of facilities, or loses edges to one clique.*

Proof: We need to determine how overlapping events can change $G(z)$ at a critical value z. To this end, we define one more graph, $H(z)$, which has one node per client, one node per facility opened in phase I, and an edge between every client j and facility i such that $w_{ij} > 0$. Thus, the edges of $G(z)$ connect facilities for which there exists a two-hop path in $H(z)$. We prove that, at a critical value of z, $H(z)$ can change only

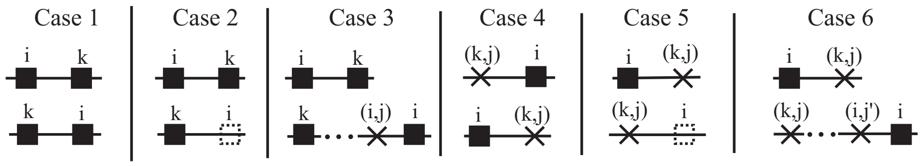

Fig. 3. Trace change cases.

by addition or deletion of one facility (along with its incident edges), or by addition or deletion of a single client-facility edge. The theorem follows.

Given the order of events, we determine the edges of $H(z)$ as follows. For each client j and open facility i, $H(z)$ includes an edge if the edge event (i, j) occurred strictly before the facility event i. Since each v_j increases at unit rate from time $t = 0$, then stops when the client j is connected, the edge event for (i, j) will either occur at $t = c_{ij}$, or not at all if the client is connected before that time. Facility events, on the other hand, change position depending on z. However, if there is a facility event for a certain value of z, that event will disappear as z changes only if it gets moved past the time where all clients are connected. Thus, the graph $H(z)$ changes in a restricted way. The vertex set changes only if a facility event is added or removed from the trace. The presence of edge (i, j) changes only if facility event i and edge event (i, j) change their relative order. Critical values of z fall into several cases, as shown in Figure 3. For ease of exposition, we refer to the top trace occurring "before" the change in z, and the bottom trace "after."

Case 1: Facilities i and k swap places. This can happen if different numbers of clients are contributing to the two facilities, causing different rates of payment. Here, the set of open facilities remains the same, and the positions of edge events relative to i and k remain the same, so $H(z)$ does not change.

Case 2: Facility i disappears when k opens first. This happens if all clients that were paying for i connect to k when it opens, and no other clients go tight to i before the end of phase I, so i remains unopened. The relative order of events remains intact, except that i is removed, so $H(z)$ changes by removal of i and all incident edges.

Case 3: Facility i jumps later in time when k opens first. Similar to case 2, this happens if all clients that were paying for i instead connect to k when it opens, causing i to remain closed for a period of time, until the next client j grows its dual enough to go tight and finish paying for i, possibly much later in the trace. Here, $H(z)$ gets one new edge (i, j).

Case 4: Facility i moves across edge (k, j). If $i = k$, then the order of the two events determines whether j has strictly positive cost share to i. Thus, as the facility event moves to the left, $H(z)$ loses the edge (i, j). If $i \neq k$, then $H(z)$ does not change, because the order of the edge event (k, j) and the facility event k (if it exists) is preserved.

Case 5: Facility i disappears as it crosses edge event (k, j) to the right (where $k \neq i$). Similar to case 2, this happens if j is the only client contributing to i, but stops when it connects to an open facility k. As in case 2, i gets deleted from $H(z)$.

Case 6: Facility i jumps later in time when the edge event (k, j) occurs before it ($k \neq i$). Similar to case 3, this happens if j is the only client contributing to i, but stops when it connects to k. However, i is opened later as some other client j' becomes tight and pays for the excess. Here, $H(z)$ gets one new edge (i, j'). ■

Clearly, the types of graph perturbations described in Theorem 5 change $\mathcal{I}(G(z))$ by at most one, which proves Theorem 4. By definition, non-degenerate k-median instances are ones where we can apply Theorem 4 at every critical value, so our algorithm is continuous when applied to these instances.

Attaining non-degeneracy.. Our last task is to prove Theorem 3. Our approach is to view UFL instances (c, z) with uniform facility costs z as points in $\mathbb{R}_+^{\mathcal{F}\times\mathcal{C}} \times \mathbb{R}_+$, i.e., the positive orthant of $(N+1)$-dimensional space, where $N = |\mathcal{F}| \cdot |\mathcal{C}|$. Each possible trace corresponds to a region of space consisting of the UFL instances that result in this trace. A k-median instance with cost vector c is represented by the ray $\{(c, z) : z > 0\}$. As long as this ray passes through no degenerate UFL points, then the k-median instance c is non-degenerate. In other words, the set of all degenerate k-median instances is simply the projection onto the $z = 0$ plane of the set of all degenerate UFL instances. Theorem 3 relies on the following result.

Theorem 6 *Each possible trace corresponds to a region of* (c, z)*-space bounded by a finite number of hyperplanes.*

We include detailed proofs of Theorems 6 and 3 in the full version of this paper. The crux is that every degenerate UFL instance lies at the intersection of two hyperplanes, hence on one of a finite number of $(N-1)$-dimensional planes. The same goes for the projection, which thus has zero Lebesgue measure in $\mathbb{R}_+^N$.

3 Facility Location Algorithm of Mettu and Plaxton

So far we have been considering the UFL algorithm of Jain & Vazirani because it has the LMP property. We now turn to a similar algorithm proposed by Mettu & Plaxton (MP). In its original form, it does not have the LMP property. However, using an LP-based analysis, we show that a slightly modified version of this algorithm attains the LMP property while delivering the same approximation factor.

Algorithm Description. The MP algorithm associates a ball of clients with each facility, and then chooses facilities in a greedy fashion, while preventing overlapping balls. Define the radii $r_i : i \in \mathcal{F}$, so that $f_i = \sum_{j\in\mathcal{C}} \max(0, r_i - c_{ij})$. Intuitively, these radii represent a sharing of the facility cost among clients. If each client in a ball of radius r_i around facility i pays a total of r_i, that will pay for the connection costs in the ball, as well as the facility cost f_i. Without loss of generality, let $r_1 \leq r_2 \leq \cdots \leq r_n$. Let B_i be the ball of radius r_i around i. In ascending order of radius, include i in the set of open facilities if there are no facilities within $2r_i$ already open. The algorithm ensures that the balls around open facilities are disjoint, so no client lies in the balls of two different open facilities. Thus, each client contributes to at most one facility cost.

Proof of Approximation Factor. We now use LP duality to prove that MP achieves an approximation factor of 3. We also prove that a slightly modified algorithm MP-β has the LMP property. The algorithm MP-β is MP with one modification. In MP-β, choose the radii r_i so that $\beta f_i = \sum_{j \in \mathcal{C}} \max(0, r_i - c_{ij})$. Our analysis uses the same LP formulation as before.

Theorem 7 *MP-β delivers a 3-approximate solution to the facility location problem for $1 \leq \beta \leq \frac{3}{2}$. Furthermore, if F is the facility cost of the algorithm's solution, C is the algorithm's connection cost, and OPT is the optimal solution cost, then $C+2\beta F \leq \sum_j v_j \leq 3OPT$.*

We prove this result by exhibiting a particular feasible dual solution. Let Z be the set of facilities opened by MP-β, and let $r_i : i \in \mathcal{F}$ be the radii used. We need to construct a set of v_j and w_{ij} from this solution. Set $w_{ij} = \frac{1}{\beta}\max(0, r_i - c_{ij})$ for $ij \in \mathcal{F}\times\mathcal{C}$. Say that j *contributes* to i if $w_{ij} > 0$. Then, set $v_j = \min_{i \in \mathcal{F}} c_{ij} + w_{ij}$. It is clear that the v and w vectors are non-negative. By the choice of the vector v, we automatically satisfy $v_j - w_{ij} \leq c_{ij}, \forall ij \in \mathcal{F}\times\mathcal{C}$. Finally, r_i and w_{ij} were chosen so that $\beta f_i = \sum_{j \in \mathcal{C}} \max(0, r_i - c_{ij}) = \beta \sum_{j \in \mathcal{C}} w_{ij}$. Thus, our dual solution is feasible.

It remains to be shown that $\sum_{j \in \mathcal{C}} d(j, Z) + 2\beta \sum_{i \in Z} f_i \leq 3 \sum_{j \in \mathcal{C}} v_j$. We will show that each $3v_j$ pays to connect j to some open facility i, and also pays for 2β times j's cost share (if one exists). Define $s_j = w_{ij}$ if there is an i such that i is open and $w_{ij} > 0$, and set $s_j = 0$ otherwise. Note that this is well defined because j can be in at most one open facility's ball. Since $f_i = \sum_{j \in \mathcal{C}} w_{ij}$, $\sum_{i \in Z} f_i = \sum_{j \in \mathcal{C}} s_j$. Furthermore, $\forall i \in Z, d(j, Z) \geq c_{ji}$ by definition. Thus, in order to show $3\sum_{j \in \mathcal{C}} v_j \geq 2\beta \sum_{i \in Z} f_i + \sum_{j \in \mathcal{C}} d(j, Z)$, it is enough to show that for all $j \in \mathcal{C}$ there exists $i \in Z$ such that $3v_j \geq c_{ij} + 2\beta s_j$.

Call the facility i that determines the minimum in $\min_{i \in \mathcal{F}} c_{ij} + w_{ij}$ the *bottleneck* of j. The proof of Theorem 7 relies on some case analysis, based on the bottleneck of j. Before we analyze the cases, we need four lemmas, stated here without proof.

Lemma 1. *For any facility $i \in \mathcal{F}$ and client $j \in \mathcal{C}$, $r_i \leq c_{ij} + \beta w_{ij}$.*

Lemma 2. *If $\beta \leq \frac{3}{2}$, and i is a bottleneck for j, then $3v_j \geq 2r_i$.*

Lemma 3. *If an open facility i is a bottleneck for j, then j cannot contribute to any other open facility.*

Lemma 4. *If a closed facility i is a bottleneck for j and k is the open facility that caused i to close, then $c_{kj} \leq \max(3, 2\beta)v_j$.*

Now we prove the theorem in cases, according to the bottleneck for each client j.

Proof of Theorem 7: We must show for all j that there is some open facility i such that $3v_j \geq c_{ij} + 2\beta w_{ij}$. Consider the bottleneck of an arbitrary client j.

Case 1: The bottleneck is some open facility i. By Lemma 3, we know that j cannot contribute to any other open facility. So connect j to facility i. If $c_{ij} < r_i$ then $0 < w_{ij} = s_j$ and $v_j = c_{ij} + s_j$. Thus, v_j pays exactly for connection cost and the cost share. If $c_{ij} \geq r_i$, we know that $s_j = 0$, since $w_{ij} = 0$, and j cannot contribute to any

other facility. So $v_j = c_{ij}$. Thus, v_j pays exactly for connection cost, and there is no cost share.
Case 2: The bottleneck is some closed facility i, and j does not contribute to any open facility. We know $s_j = 0$ since j does not contribute to any open facility. We also know there is some open facility k that caused i to close. Connect j to k. By Lemma 4, we know that $c_{kj} \leq \max(3, 2\beta)v_j$. Since $\beta \leq \frac{3}{2}$, we have that $3v_j \geq c_{kj}$. Thus, $3v_j$ pays for the connection cost, and there is no cost share.
Case 3: The bottleneck is some closed facility i, and there is some open facility l with $w_{lj} > 0$, and l was not the reason that i closed. Since $w_{lj} > 0$, $s_j = w_{lj}$. Connect j to l incurring $c_{lj} + w_{lj}$. Since $w_{lj} = s_j$, we have that $c_{lj} + \beta s_j = r_l$. Since k and l are both open, we have that $c_{lk} \geq 2r_l$. Using the triangle inequality, this gives $2c_{lj} + 2\beta s_j \leq c_{lk} \leq c_{lj} + c_{kj}$, or $c_{lj} + 2\beta s_j \leq c_{kj}$. Just as in Case 2, we know there is some open facility $k \neq l$ that prevented i from opening, which means $c_{ik} \leq 2r_i$. By Lemma 4, we know $c_{kj} \leq 3v_j$. So, putting it all together, we have $c_{lj} + 2\beta s_j \leq c_{kj} \leq 3v_j$. Thus, $3v_j$ pays for the connection cost, and 2β times the cost share.
Case 4: The bottleneck is some closed facility i and there is some open facility k with $w_{kj} > 0$ and k caused i to be closed. Here, $s_j = w_{kj}$. From Lemma 2, we know that $3v_j \geq 2r_i$. Since k caused i to close, $r_i \geq r_k = c_{kj} + \beta s_j$. Thus, we have $3v_j \geq 2r_i \geq 2c_{kj} + 2\beta s_j \geq c_{kj} + 2\beta s_j$. So $3v_j$ pays for the connection cost and 2β times the cost share.

Thus, in each case, we have shown that there is an open facility i that satisfies $3v_j \geq c_{ij} + 2\beta s_j$ which shows that the algorithm delivers a solution that satisfies $C + 2\beta F \leq 3OPT$, giving a 3-approximation so long as $\beta \geq \frac{1}{2}$. ∎

4 Final thoughts

The preceding theorem shows that the algorithm MP-$\frac{3}{2}$ has the LMP property necessary to build a k-median algorithm. The primary benefit of using MP-$\frac{3}{2}$ instead of another LMP algorithm with guarantee 3 is the running time. The k-median approximation algorithm runs the facility location algorithm several times as a black box. Whereas the original JV facility location algorithm had a running time of $O(|\mathcal{F}||\mathcal{C}| \log |\mathcal{F}||\mathcal{C}|)$, the algorithm MP-$\frac{3}{2}$ can be implemented to run in $O(|\mathcal{F}|^2 + |\mathcal{F}||\mathcal{C}|)$ time.

Any LMP algorithm with guarantee c that also has the continuity property analogous to Theorem 4 immediately yields a c-approximation for the k-median problem, because we can simply search for a value of z for which we open exactly k facilities. Unfortunately MP-$\frac{3}{2}$ is not continuous. We include an example demonstrating this fact in the full version of this paper. The tightest LMP result is the dual fitting algorithm of [17], which yields a factor of 2. However, on the star instance of Figure 1, this algorithm jumps from opening 1 facility to opening h of them at $z = \frac{h+\epsilon}{h-1}$. Thus, our modification of JV is the only LMP algorithm so far that has this property.

An important direction for future research is to identify a rule for computing maximal independent sets in polynomial time that satisfy the continuity property of Theorem 4, with $\mathcal{I}(G(z))$ replaced by $|S(z)|$. This would convert our existential result into a polynomial time 3-approximation algorithm for the k-median problem.

One algorithmic consequence of Theorem 3 is that we can always make an arbitrarily small perturbation to our given instance to transform it into a non-degenerate instance. For purposes of applying Theorems 4 and 5, it suffices to process trace changes one at a time for degenerate values of z.

References

1. A. Ageev & M. Sviridenko. An approximation algorithm for the uncapacitated facility location problem. Manuscript, 1997.
2. S. Arora, B. Bollobas, & L. Lovász. Proving integrality gaps without knowing the linear program. *43rd FOCS*, 313–322, 2002.
3. V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala, & V. Pandit. Local search heuristic for k-median and facility location problems. *33rd STOC*, 21–29, 2001.
4. M. Balinski. On finding integer solutions to linear programs. In *Proc. IBM Scientific Computing Symp. on Combinatorial Problems*, 225–248, 1966.
5. Y. Bartal. On approximating arbitrary metrics by tree metrics. *30th STOC*, 161–168, 1998.
6. Y. Bartal. Probabilistic approximations of metric spaces and its algorithmic applications. *37th FOCS*, 184–193, 1996.
7. M. Charikar & S. Guha. Improved combinatorial algorithms for the facility location and k-median problems. *40th FOCS*, 378–388, 1999.
8. M. Charikar, S. Guha, E. Tardos, & D.B. Shmoys. A constant-factor approximation algorithm for the k-median problem. *31st STOC*, 1–10, 1999.
9. F. Chudak & D.B. Shmoys. Improved approximation algorithms for uncapacitated facility location. *SIAM J. Comput.*, to appear.
10. S. Guha & S. Khuller. Greedy strikes back: improved facility location algorithms. *J. Alg. 31*, 228–248, 1999.
11. D. Hochbaum. Heuristics for the fixed cost median problem. *Math. Prog. 22*, 148–162, 1982.
12. K. Jain & V. Vazirani. Approximation algorithms for metric facility location and k-median problems using primal-dual schema and Lagrangian relaxation. *JACM 48*, 274–296, 2001.
13. K. Jain & V. Vazirani. Primal-dual approximation algorithms for metric facility location and k-median problems. *40th FOCS*, 2–13, 1999.
14. M. Korupolu, C.G. Plaxton, & R. Rajaraman. Analysis of a local search heuristic for facility location problems. *J. Alg. 37*, 146–188, 2000.
15. J.H. Lin & J. Vitter. Approximation algorithms for geometric median problems. *IPL 44*, 245–249, 1992.
16. M. Mahdian, Y. Ye, & J. Zhang. Improved approximation algorithms for metric facility location problems. *4th APPROX, LNCS 2462*, 229–242, 2002.
authors Optimization:
17. K. Jain, M. Mahdian, E. Markakis, A. Saberi, & V. Vazirani. Greedy facility location algorithms analyzed using dual fitting with factor-revealing LP To appear in *JACM*
18. N. Meggido. Combinatorial optimization with rational objective functions. *Math. OR*, 4:414–424, 1979.
19. R. Mettu & C.G. Plaxton. The online median problem. *41st FOCS*, 339–348, 2000.
20. M. Pál & Éva Tardos. Strategy proof mechanisms via primal-dual algorithms. To appear in *44th FOCS*, 2003.
21. D.B. Shmoys, É. Tardos, & K. Aardal. Approximation algorithms for facility location problems. *29th STOC*, 265–274, 1997.
22. M. Sviridenko. An improved approximation algorithm for the metric uncapacitated facility location problem. *9th IPCO, LNCS 2337*, 240–257, 2002.