Application of computational geometry to network p-center location problems

Binay Bhattacharya *† Qiaosheng Shi †

Abstract

In this paper we showed that a $p(\geq 2)$-center location problem in general networks can be transformed to the well known Klee's measure problem [5]. This resulted in an improved algorithm for the continuous case with running time $O(m^p n^{p/2} 2^{\log^* n} \log n)$. The previous best result for the problem is $O(m^p n^p \alpha(n) \log n)$ where $\alpha(n)$ is the inverse Ackermann function [16]. When the underlying network is a partial k-tree (k fixed), by exploiting the geometry inherent in the problem we showed that the discrete p-center problem can be solved in $O(pn^p \log^k n)$ time.

1 Introduction

The *network p-center problem* is defined on a weighted undirected network $G = (V(G), E(G))$, where each vertex $v \in V(G)$ has a non-negative weight $w(v)$ and each edge $e \in E(G)$ has a positive length $l(e)$. Let $A(G)$ denote the continuum set of points on the edges of G. For $x, y \in A(G)$, $\pi(x, y)$ denote the *shortest path* in G from x to y, and $d(x, y)$ denote the length of $\pi(x, y)$. Let $\mathcal{D}(G)$ be the set of demand points (or the *demand set*) and $\mathcal{X}(G)$ be the set of candidate facility locations in G. In a p-center problem, a set X of p centers is to be located in $\mathcal{X}(G)$ so that the maximum (weighted) distance from a demand point in $\mathcal{D}(G)$ to its nearest center in X is minimized, i.e.,

$$\min_{X \subseteq \mathcal{X}(G), |X| = p} \{F(X, \mathcal{D}(G)) = \max_{y \in \mathcal{D}(G)} \{w(y) \cdot d(y, X)\}\}.$$

Here $d(y, X) = \min_{x \in X} d(y, x)$ and $F(X, \mathcal{D}(G))$ denotes the cost of serving the demand set $\mathcal{D}(G)$ using facilities in X.

A value $r > 0$ is *feasible* if there exists a set X of at most p points (centers) in $\mathcal{X}(G)$ such that the distance between any demand point in $\mathcal{D}(G)$ and its nearest center in X is not greater than r. An approach to test whether a given positive value r is feasible is called a *feasibility test*.

Our study in this paper is restricted to the case where $\mathcal{D}(G) = V(G)$. When p centers are restricted to be vertices of G, we call it a *discrete problem*. The network p-center problem is called *continuous* when $\mathcal{X}(G) = A(G)$. The continuous/discrete network p-center problems have been shown to be NP-hard in general networks [9]. But, center problems are no longer NP-hard when either p is constant [9, 13, 16], or the underlying network is restricted to be a specialized network, such as a tree [9], a cactus [1], or a partial k-tree (fixed k) [10]. In the paper we study p-center problems in general networks and partial k-trees (k fixed) for a constant p and provide improved algorithms by exploiting the geometric properties of the problems.

*Research was partially supported by MITACS and NSERC

†School of Computing Science, Simon Fraser University, Burnaby B.C., Canada. V5A 1S6, {binay, qshi1}@cs.sfu.ca

2 Continuous p-center problem in general networks

The best known algorithms [9, 13, 16] to solve the continuous p-center problem in a general network are based on the following two simple observations.

Observation 1 *[9] There exists a p-center solution such that all the centers are intersection points of service functions of pairs of vertices on edges and therefore, the optimal objective value is of the form $(w(u) \cdot w(v) \cdot L(u, v))/(w(u) + w(v))$, where $L(u, v)$ is the length of the shortest path connecting u and v through edge e for some pair of vertices $u, v \in V(G)$.*

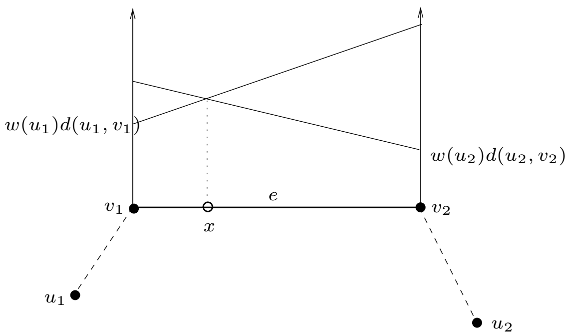

Figure 1: Observation 1.

Refer to Figure 1. The intersection point x of service functions of vertices u_1 and u_2 on edge e is a candidate location of facilities. Also, $w(u_1)d(u_1, x) = w(u_2)d(u_2, x)$ is a candidate value for the optimal objective value. Therefore, there are at most $O(n^2)$ candidate points on each edge e where centers may be located in an optimal solution. Also, each candidate point determines a candidate optimal cost. Let $\mathcal{R}$ denote the set of $O(mn^2)$ candidate values. It is not hard to see that the set of $O(mn^2)$ candidate points and the set $\mathcal{R}$ can be computed in $O(mn^2)$ time. Based on Observation 1 and an $O(mn \log n)$-time algorithm for the continuous 1-center problem, Kariv and Hakimi [9] proposed an $O(m^p n^{2p-1} \log n/(p-1)!)$-time algorithm for finding an optimal p-center.

The second observation is for feasibility tests of the continuous p-center problem.

Observation 2 *[13] If r is feasible, then there is a p-center solution in which each center is located at a (weighted) distance of exactly r from some vertex and all vertices are covered with service cost $\leq r$.*

The advantage of Observation 2 is that only $O(mn)$ candidate points are needed to be considered for a feasibility test. Based on this property, Moreno [13] proposed an $O(m^p n^{p+1})$-time algorithm to test the feasibility of a given value r. Since the optimal service cost, denoted by r_p, is an element of a set $\mathcal{R}$ (Observation 1), r_p can be found by performing $O(\log n)$ feasibility tests.

Tamir [16] improved Moreno's result [13] by efficiently solving a feasibility test for the 2-center problem in $O(m^2 n^2 \alpha(n))$ time. Here $\alpha(n)$ is the inverse Ackermann function [8]. The improvements [16] for $p \geq 3$ are achieved as follows. To determine the feasibility of r for the p-center problem, a set of $p-2$ points is selected from the $O(mn)$ candidate points, and then the 2-center feasibility test is applied to the vertices that are not covered by the selected $p-2$ centers within (weighted) distance r. Therefore, the $O(m^p n^p \alpha(n) \log n)$ bound is achieved for the continuous p-center problems.

Next we present an improved algorithm for the continuous p-center problem in general networks.

2.1 Main idea and overall approach

To achieve a better upper bound, we continue to decrease the size of the set that contains an optimal p-center. Observation 3 shows that instead of $O(mn)$ candidate points, only m candidate continuous regions (i.e., edges) and n candidate points (i.e., vertices) are considered for a feasibility test.

Observation 3 *If r is feasible, then there is a p-center solution in which every edge (not including its two endpoints) contains at most one center and all vertices are covered with service cost $\leq r$.*

A *local feasibility test* of r is to determine if there exists a set of p centers on a given set $E_{p'}$ of p' $(0 \leq p' \leq p)$ edges $\{e_1, \ldots, e_{p'}\}$ (note that each edge contains one center and does not include its two endpoints) and a given set of $p-p'$ vertices such that all vertices can be served within r. It is easy to see that the feasibility test of r can be completed by solving $O((m+n)^p) = O(m^p)$ local feasibility tests of r on all possible subsets of p' edges and $p-p'$ vertices, $0 \leq p' \leq p$.

Our algorithm is described as follows.

Step 1: Compute $\mathcal{R}$ that contains the optimal cost r_p.

Step 2: Perform a binary search over $\mathcal{R}$. At each iteration, test the feasibility of a non-negative value r as follows. For each set $E_{p'} \subseteq E(G)$ of p' edges and each set of $p-p'$ vertices, $0 \leq p' \leq p$,

- **Step 2.1:** remove all vertices that can be covered by $p-p'$ vertices with service cost $\leq r$; and
- **Step 2.2:** for the remaining vertices, execute the local feasibility test of r on the set $E_{p'}$ as described in the remaining part of this section.

It is sufficient to show our approach for a local feasibility test of r on a set E_p of p edges. The main idea is to transform the local feasibility test of r on E_p to a general geometrical problem called p-dimensional *Klee's measure problem* (for short, KMP) [5, 14].

Definition 1 (Klee's Measure Problem) *Given a set of intervals (of the real line), find the length of their union.*

The natural extension of KMP to d-dimensional space is to ask for the d-dimensional measure of a set of d-boxes. A *d-box* is the cartesian product of d intervals in d-dimensional space. It is known that, given a set of n d-boxes, a $d(\geq 2)$-dimensional KMP can be solved in time $O(n^{d/2} \log n)$ [14], which can be improved to $O(n^{d/2} 2^{\log^* n})$ [5] ($\log^* n$ denotes the iterated logarithm of n). Thus, a feasibility test can be solved in $O(m^p n^{p/2} 2^{\log^* n})$ time if we are able to transform a local feasibility test into a KMP. The following theorem is then implied.

Theorem 2 *The continuous p-center problems, for $p \geq 2$, can be solved in $O(m^p n^{p/2} 2^{\log^* n} \log n)$ time.*

In the remaining part of this section, we describe the process of transforming a local feasibility test to a p-dimensional KMP.

2.2 Transformation of a local feasibility test to a p-dimensional KMP

Let us consider the case where $p = 2$. The transformation for the case where $p > 2$ can be developed in a similar way. Let $e_1 : \overline{u_1 v_1}$ and $e_2 : \overline{u_2 v_2}$ be the two edges to test the local feasibility of r. A local 2-center solution is composed of two points (not vertices) in which one point lies on e_1 and the other one lies on e_2.

We consider a 2-dimensional space in which x_i-axis represents edge $e_i, i = 1, 2$. Let u_1 and u_2 be the origin,

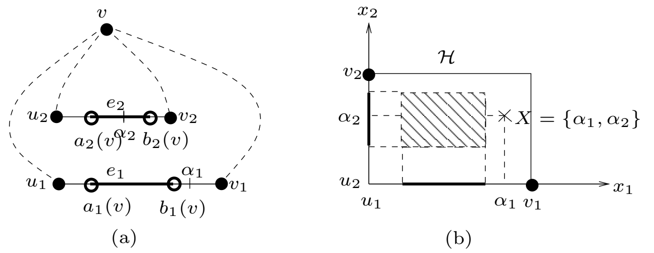

Figure 2: Mapping a 2-center local feasibility test to a 2-dimensional KMP.

as shown in Figure 2(b). In this coordinate system, the x_i-coordinate of a point represents a location on edge e_i with respect to $u_i, i = 1, 2$. Therefore, a point in this 2-dimensional space can be considered as a possible 2-center solution on edges e_1, e_2. We denote a point y by $(x_1(y), x_2(y))$. Clearly, only points within the bounded rectangular area $\mathcal{H} : \{y|0 < x_1(y) < l(e_1), 0 < x_2(y) < l(e_2)\}$ are candidate 2-center solutions on e_1, e_2.

For a vertex v, there is at most one continuous region on each edge $e_i, i = 1, 2$, denoted by $R_i(v)$, which contains all points on e_i with (weighted) distance to v larger than r. It is possible that $R_i(v)$ is empty for some i ($\in \{1, 2\}$), in which case v can be served by any 2-center solution on e_1, e_2 with service cost $\leq r$. In Figure 2(a), the bold (partial) edge of e_1 (resp. e_2) is $R_1(v)$ (resp. $R_2(v)$). Let $a_i(v)$ (resp. $b_i(v)$) be the left (resp. right) endpoint of $R_i(v), i = 1, 2$. Note that $R_i(v), i = 1, 2$ must be an open region.

A rectangular area in the 2-dimensional space (the shadow part in Figure 2(b)) is obtained for every demand vertex v, denoted by $\mathcal{H}(v)$, which is constructed from the two continuous regions $R_1(v), R_2(v)$. That is, $\mathcal{H}(v) = \{y|x_1(y) \in R_1(v), x_2(y) \in R_2(v)\}$. It is easy to see that any 2-center solution (point) in $\mathcal{H}(v)$ cannot cover v with a service cost $\leq r$ and any 2-center solution in $\mathcal{H} \setminus \mathcal{H}(v)$ can cover v with a service cost $\leq r$. In Figure 2, the 2-center solution $X = \{\alpha_1, \alpha_2\}$ can cover v with a service cost $\leq r$, but any solution in the shadow area cannot. We call $\mathcal{H}(v)$ the *forbidden area* of v. Note that, the boundary of $\mathcal{H}(v)$ is not included in $\mathcal{H}(v)$.

We compute such forbidden areas for all remaining vertices. Thus, the local feasibility test on edges e_1, e_2 is transformed into the following question: *does the union* $\bigcup_{v \in V(G)} \mathcal{H}(v)$ *of forbidden areas cover* $\mathcal{H}$*?* If the answer is 'yes' then r is infeasible on edges e_1, e_2, otherwise r is feasible. This question can be answered by solving a 2-dimensional KMP on a new set of rectangles, which are constructed from these forbidden areas. We have to be careful since the boundary of a forbidden area is not forbidden. This is handled by appropriately shrinking/expanding the boundary appropriately.

Lemma 3 shows that we can construct a set of rectangles in a way such that the above question can be answered by solving a KMP. Let

$$\epsilon_1 = \min_{u,v \in V(G), i=1,2} \{|a_i(u) - a_i(v)| : \ a_i(u) \neq a_i(v)\},$$

$$\epsilon_2 = \min_{u,v \in V(G), i=1,2} \{|a_i(u) - b_i(v)| : \ a_i(u) \neq b_i(v)\},$$

and

$$\epsilon_3 = \min_{u,v \in V(G), i=1,2} \{|b_i(u) - b_i(v)| : \ b_i(u) \neq b_i(v)\}.$$

Let ϵ be $(\min\{\epsilon_1, \epsilon_2, \epsilon_3\})/2$. Clearly, $\epsilon > 0$ and ϵ can be computed in $O(n \log n)$ time.

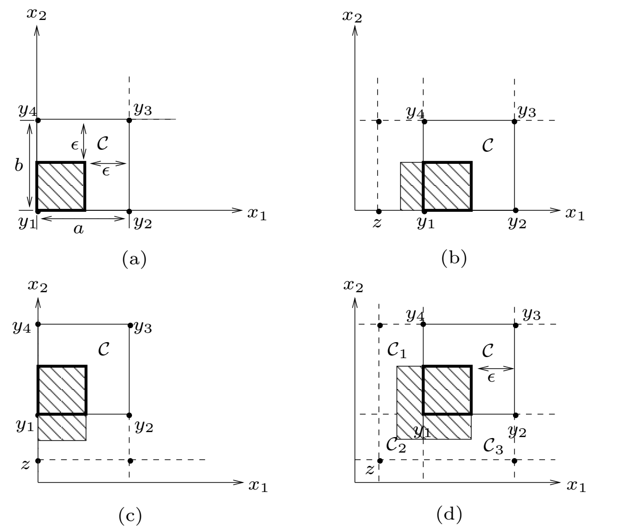

Figure 3: Lemma 3 (a) Case 1: y_1 is located at the origin; (b) Case 2: $x_2(y_1) = 0$; (c) Case 3: $x_1(y_1) = 0$; (d) Case 4: $x_1(y_1) \neq 0$ and $x_2(y_1) \neq 0$.

Lemma 3 *For each* $v \in V(G)$*, let* $a'_i(v) = a_i(v) + \epsilon$ *if* $a_i(v) \neq 0$*; let* $b'_i(v) = b_i(v) - \epsilon$ *if* $b_i(v) \neq l(e_i)$*,* $i = 1, 2$*. Let* $\mathcal{H}'(v) = \{y|a'_1(v) \leq x_1(y) \leq b'_1(v), a'_2(v) \leq x_2(y) \leq b'_2(v)\}$*.*

Therefore, the measure of $\sum_{v \in V(G)} \mathcal{H}'(v)$ *is equal to the measure of* $\mathcal{H}$ *if and only if the union* $\bigcup_{v \in V(G)} \mathcal{H}(v)$ *covers* $\mathcal{H}$*.*

Proof. It is trivial that $\mathcal{H}'(v) \subseteq \mathcal{H}(v), v \in V(G)$. Hence, if the measure of $\sum_{v \in V(G)} \mathcal{H}'(v)$ is equal to the measure of $\mathcal{H}$ then the union $\bigcup_{v \in V(G)} \mathcal{H}(v)$ covers $\mathcal{H}$. We prove in the following that if $\sum_{v \in V(G)} \mathcal{H}(v)$ covers $\mathcal{H}$, then the measure of $\sum_{v \in V(G)} \mathcal{H}'(v)$ is equal to the measure of $\mathcal{H}$.

We draw lines along boundaries of all forbidden areas $\mathcal{H}(v), v \in V(G)$. These lines partition $\mathcal{H}$ into rectangular cells. From the computation of ϵ, it is evident that the vertical (resp. horizontal) length of each cell is at least 2ϵ.

Refer to Figure 3. Let y_1, y_2, y_3, y_4 be the four corner points of a cell $\mathcal{C}$. Let a be the horizontal length of $\mathcal{C}$ and

b be its vertical length. Note that $a \geq 2\epsilon$ and $b \geq 2\epsilon$. We want to show that there exists a new rectangle $\mathcal{H}'(v)$ such that y_1 is contained by $\mathcal{H}'(v)$, and such that all the points z in cell $\mathcal{C}$ with $x_1(z) \leq x_1(y_1) + a - \epsilon$ and $x_2(z) \leq x_2(y_1) + b - \epsilon$ lie within $\mathcal{H}'(v)$. That is, the shadow part in $\mathcal{C}$ is contained by a new rectangle $\mathcal{H}'(v)$.

We have four possible cases according to locations of y_1, as shown in Figure 3. We only consider Case 4, the other cases can be handled analogously. In Case 4, y_1 is not on the boundary of $\mathcal{H}$. As we know, for a forbidden area $\mathcal{H}(v)$, any point that is on the boundary of $\mathcal{H}(v)$ but not on the boundary of $\mathcal{H}$, is not contained in $\mathcal{H}(v)$. Thus, a forbidden area that contains y_1 must cover the interior part of cell $\mathcal{C}$ and the interior parts of cells $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3$ where $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3$ are cells adjacent to $\mathcal{C}$, as shown in Figure 3(d). Note that the vertical (resp. horizontal) length of each cell is at least 2ϵ. Therefore, the new rectangle constructed from this forbidden area must cover the shadow parts in cells $\mathcal{C}, \mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3$, as desired.

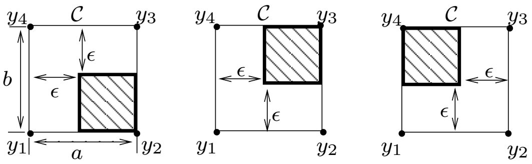

Figure 4: Lemma 3: (a) y_2; (b) y_3; (c) y_4.

Similarly, the above statement about y_1 is also correct for y_2, y_3, and y_4 (see Figure 4). Observe that the union of these four shadow rectangles cover the cell $\mathcal{C}$ (including the boundary of $\mathcal{C}$). Therefore, the measure of $\sum_{v \in V(G)} \mathcal{H}'(v)$ is equal to the measure of $\mathcal{H}$, which completes the proof of this lemma. □

It takes $O(n \log n)$ time to compute the new set of rectangles $\mathcal{H}'(v)$ from $\mathcal{H}(v), v \in V(G)$. Therefore, a local feasibility test on edges e_1, e_2 can be solved in $O(n \log n)$ time.

The extension of the above approach to the case when $p > 2$ is straightforward. Now a local p-center solution is represented as a point in a p-dimensional box (p-box) $\mathcal{H}$ and for each demand vertex v, we obtain a p-box in $\mathcal{H}$ containing all p-center solutions that serve v with a service cost $> r$. Thus, the local feasibility test on edges $e_1, \ldots, e_p$, is transformed into the following p-dimensional Klee's measure problem: *does the union of $O(n)$ axis-parallel p-boxes cover $\mathcal{H}$?* Therefore, we have the following lemma.

Lemma 4 *A local feasibility test of the weighted continuous p-center problem on p edges $e_1, \cdots, e_p$ can be solved in $O(n^{p/2} 2^{\log^* n})$ time, for $p > 1$.*

This establishes Theorem 2.

3 Discrete p-center problems in a partial k-tree

A *tree decomposition* of a network $G = (V(G), E(G))$ is a pair $(\{B_i : i \in I\}, \mathcal{T} = (I, Y))$ with the following properties [3]:

(i) $\bigcup_{i \in I} B_i = V(G)$;

(ii) For every edge $e = (v, w) \in E(G)$, there is an $i \in I$ with $v, w \in B_i$;

(iii) For all $i, j, q \in I$, if j lies on the path between i and q in $\mathcal{T}$, then $B_i \cap B_q \subseteq B_j$.

The *treewidth* of $(\{B_i : i \in I\}, \mathcal{T})$ is $\max_{i \in I} |B_i| - 1$. The treewidth of G is the minimum treewidth over all possible tree decompositions. A *partial k-tree* is a graph whose treewidth is k. It is known that a tree decomposition (of treewidth k) of a partial k-tree G (fixed k), denoted by $\mathcal{TD}(G)$, can be found in linear time [3].

Given a tree decomposition $\mathcal{TD}(G)$ of treewidth k, an $O(p^2 n^{k+2})$ algorithm [10] was proposed to solve the discrete p-center problems, which is based on the dynamic programming technique.

In fact, the approach of Granot and Skorin-Kapov [10] can be adopted to solve the continuous p-center problem by combining the result from Observation 2. Recall that if a non-negative value r is feasible, then there is a p-center solution in which each center is located at a (weighted) distance of exactly r from some demand vertex, and all demand vertices are covered with service cost $\leq r$ (Observation 2). Therefore, only $O(kn^2)$ candidate points are needed to be considered for the feasibility test of r. Also, since the optimal service cost is in a set of size $O(n^3)$ (Observation 1), the optimal service cost can be found by performing $O(\log n)$ feasibility tests.

Theorem 5 *Given a tree decomposition (of treewidth k) of a partial k-tree, the continuous p-center problem can be solved in $O(p^2 n^{2k+3} \log n)$ time.*

In this section, we present an $O(pn^p \log^k n)$-time algorithm for the discrete p-center problem in a partial k-tree when p is small. Note that the discrete p-center problem, $p \geq 2$, in a general network is trivially solvable in $O(pn^{p+1})$ time by testing all possible solutions.

3.1 An $O(pn^p \log^k n)$-time algorithm

A distance query of a pair of points x, y in a network is to obtain the distance between x, y. Considering the tight relationship between the service cost and distance queries, an efficient approach is to preprocess the network so that distance queries can be efficiently answered. This approach is particularly promising when the network is sparse [6]. Chaudhuri and Zaroliagis [6] gave algorithms for distance queries that depend on the treewidth of the input network. Their algorithms can answer each distance query in $O(1)$ time for constant treewidth networks after $O(n \log n)$ preprocessing. Based on this result, we introduced a two-level tree decomposition structure [1] on a partial k-tree network, which can be built on any partial k-tree G in $O(n \log^2 n)$ time requiring $O(n \log^2 n)$ storage space for $k = 2$ and in $O(nk \log^{k-1} n)$ time requiring $O(nk \log^{k-1} n)$ storage space for $k > 2$. Given such a two-level tree decomposition structure, the service cost of any set X of p centers to the demand set $V(G)$, i.e., $F(X, V(G))$, can be answered in time $O(p \log^2 n)$ for $k = 2$ and in time $O(pk \log^{k-1} n)$ for $k > 2$. The main idea behind this two-level tree decomposition data structure is briefly described below for the case when G is a partial 2-tree. Recently similar idea has been used in [4] to compute some graph properties.

Given a subgraph G' represented by a subtree of $\mathcal{TD}(G)$ and a point x outside G', there is a 2-separator in G', say $\{u_1, u_2\}$, between G' and x. The service cost of x to cover $v \in V(G')$ is $w(v) \cdot \min\{d(v, u_1) + d(u_1, x), d(v, u_2) + d(u_2, x)\}$. Let $a = d(x, u_1) - d(x, u_2)$ and $a' = d(v, u_1) - d(v, u_2)$. Clearly the shortest path $\pi(v, x)$ goes through u_1 if $a + a' < 0$, otherwise $\pi(v, x)$ goes through u_2.

Based on the above observation, we create two lists of the vertices in G', $\mathcal{J}_1$ and $\mathcal{J}_2$. The vertices of $\mathcal{J}_1$ are sorted in the increasing order of $\chi_1(\cdot)$ where $\chi_1(v)$ is the distance difference from vertex $v(\in V(G'))$ to the 2-separator $\{u_1, u_2\}$, i.e., $\chi_1(v) = d(v, u_1) - d(v, u_2)$. The vertices of $\mathcal{J}_2$ are sorted in the increasing order of $\chi_2(\cdot)$ where $\chi_2(v) = d(v, u_2) - d(v, u_1)$ for all $v \in V(G')$. These two lists $\mathcal{J}_1$ and $\mathcal{J}_2$ are associated with u_1 and u_2 respectively.

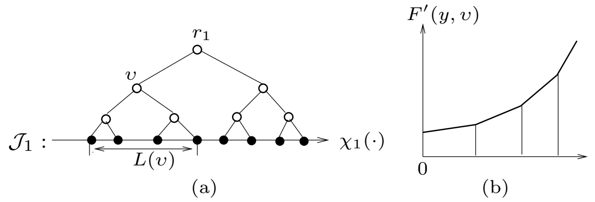

Figure 5: A balanced binary search tree T_1 over χ_1.

A balanced binary search tree over $\mathcal{J}_1$ (resp. $\mathcal{J}_2$) is built (see Figure 5(a)), denoted by T_1 (resp. T_2). Let r_1 and r_2 be the root nodes of T_1 and T_2 respectively. The set of vertices in G' whose shortest paths to a point x outside G' go through u_1 (resp. u_2) is represented by $O(\log |V(G')|)$ sublists in $\mathcal{J}_1$ (resp. $\mathcal{J}_2$). It is not hard to see that, these sublists can be identified in $O(\log |V(G')|)$ time.

Computing $F(x, V(G'))$ for any point x outside G' In the following we show that $F(x, V(G'))$ can be computed in $O(\log |V(G')|)$ time by applying the fractional cascading technique [7].

For every node v in T_1, let $L(v)$ be the set of leaves descending from v in T_1 (see Figure 5(a)), and let $F'(y, v)$ denote the service cost function of a point, lying outside G' with distance y to u_1, to cover the vertices in $L(v)$ via u_1. For every vertex $z \in L(v)$, there is at most one continuous region of y in which z determines the service cost $F'(y, v)$, called the *dominating region* of z in the sublist $L(v)$. Moreover, these dominating regions are sorted in increasing order of weights of vertices in $L(v)$ (see Figure 5(b)).

Therefore, in a bottom-up way, we can compute the service cost function $F'(y, v)$ for every v in T_1 in total time $O(|V(G')| \log |V(G')|)$. By applying the fractional cascading technique [7], we are able to locate the weighted farthest vertex in $L(v)$ for every v in T_1 in constant time after an $O(\log |V(G')|)$-time computation of the weighted farthest vertex in $L(r_1)$. A similar result is obtained after applying the fractional cascading technique on T_2. Therefore, the total cost of computing the service cost $F(x, V(G'))$ is $O(\log |V(G')|)$ after $O(|V(G')| \log |V(G')|)$ preprocessing time.

A two-level tree decomposition of G Since the tree decomposition $\mathcal{TD}(G)$ of G might not be balanced, we add another balanced tree structure over $\mathcal{TD}(G)$, such that the height of the new tree $\overline{\mathcal{TD}}(G)$ is logarithmic. We call such a balanced tree structure $\overline{\mathcal{TD}}(G)$ *a two-level tree decomposition of* G. There are several methods to achieve this, such as centroid tree decomposition [12], spine tree decomposition [2] etc. We can see that a two-level tree decomposition data structure of a partial 2-tree can be computed in $O(n \log^2 n)$ time requiring $O(n \log n)$ storage space and the service cost of a set of p points in the partial 2-tree can be answered in $O(p \log^2 n)$. Thus,

Theorem 6 *Given a tree decomposition (of treewidth 2) of a partial 2-tree G, the discrete p-center problem can be solved in $O(pn^p \log^2 n)$ time.*

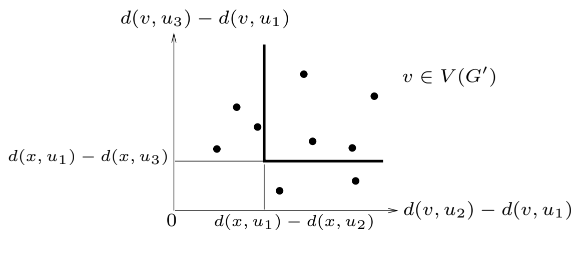

Figure 6: The set of vertices in $V(G')$ to which the shortest path from x goes through u_1.

This result can be extended to a partial k-tree, $k > 2$. In the case when G is a partial k-tree, given a subgraph

G' represented by a subtree of $\mathcal{TD}(G)$ and a point x outside G', there is a k-separator in G' between G' and x. Refer to Figure 6 in which G is a partial 3-tree. All vertices in $V(G')$ are embedded in a 2-dimensional space (note that it is a $(k-1)$-dimensional space when G is a partial k-tree). Given a point x with its distances to the 3-separator $\{u_1, u_2, u_3\}$, all the vertices lying above and right of the bold line in Figure 6 are the vertices in $V(G')$ to which the shortest path from x goes through u_1. The service cost of x to these vertices can be computed in $O(\log |V(G')|)$ time by the combining priority search tree [11] and the fractional cascading technique [7] after $O(|V(G')| \log |V(G')|)$ preprocessing time.

Similarly, when G is a partial k-tree, we can compute $F(x, V(G'))$ in $O(k \log^{k-2} |V(G')|)$ time after $O(k|V(G')| \log^{k-2} |V(G')|)$ preprocessing time. Hence, for $k > 2$, a two-level tree decomposition data structure of a partial k-tree can be computed in $O(nk \log^{k-1} n)$ time requiring $O(nk \log^{k-1} n)$ storage space such that the service cost of a set of p points in the partial k-tree can be answered in $O(pk \log^{k-1} n)$.

We have the following theorem.

Theorem 7 *Given a tree decomposition (of treewidth $k > 2$) of a partial k-tree G, the discrete p-center problem can be solved in $O(pkn^p \log^{k-1} n)$ time.*

4 Future work

For the general p-center problem in which the demand set contains all points of the underlying network (i.e., $\mathcal{D}(G) = A(G)$), a candidate set containing the optimal solution value is characterized in Tamir's paper [15]. In spite of the nice structure, the size of this set is not polynomial even for simple structures such as partial 2-trees. Until now, no efficient algorithm is known for the problem in a general network or a partial k-tree.

References

[1] B. Ben-Moshe, B.K. Bhattacharya, Q. Shi and A. Tamir, "Efficient algorithms for center problems in cactus networks", *Theor. Comput. Sci.*, 378(3):237–252, 2007.

[2] R. Benkoczi, "Cardinality constrainted facility location problems in trees", *Ph.D. Thesis, School of Computing Science, SFU, Canada*, 2004.

[3] H.L. Bodlaender, "A linear-time algorithm for finding tree-decompositions of small treewidth" *SIAM J. Comput.*, 25(6):1305–1317, 1996.

[4] S. Cabello and C. Knauer, "Algorithms for graphs of bounded treewidth via orthogonal range searching", in *EuroCG'08.*

[5] T.M. Chan, "A (slightly) faster algorithm for Klee's measure problem", in *SoCG'08.*

[6] S. Chaudhuri and C.D. Zaroliagis, "Shortest paths in digraphs of small treewidth. part I: sequential algorithms", *Algorithmica*, 27(3):212–226, 2000.

[7] B. Chazelle and L.J. Guibas, "Fractional cascading: I. a data structuring technique", *Algorithmica*, 1(2):133–162, 1986.

[8] T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein, "Introduction to algorithms", *The MIT Press and McGraw-Hill Book Company*, 2001.

[9] O. Kariv and S.L. Hakimi, "An algorithmic approach to network location problems. I: the p-centers", *SIAM Journal on Applied Mathematics*, 37(3):513–538, 1979.

[10] D. Granot and D. Skorin-Kapov, "On some optimization problems on k-trees and partial k-trees", *Disc. App. Math.*, 48(2):129–145, 1994.

[11] E.M. McCreight, "Priority search trees", *SIAM J. Comput.*, 14(2):257–276, 1985.

[12] N. Megiddo and A. Tamir, "New results on the complexity of p-center problems," *SIAM J. Comput.* 12:4 (1983) 751–758.

[13] J.A. Moreno, "A new result on the complexity of the p-center problem", *Technical Report, Universidad Complutense*, 1986.

[14] M.H. Overmars and C.K. Yap, "New upper bounds in Klee's measure problem", *SIAM J. Comput.*, 20(6):1034–1045, 1991.

[15] A. Tamir, "On the solution value of the continuous p-center location problem on a graph", *Math. Oper. Res.*, 12(2):340–349, 1987.

[16] A. Tamir, "Improved complexity bounds for center location problems on networks by using dynamic data structures", *SIAM J. Discret. Math.*, 1(3):377–396, 1988.