Kinetic Maintenance of Mobile k-Centres on Trees

Stephane Durocher[1,⋆] and Christophe Paul[2]

[1] School of Computer Science, McGill University, Montréal, Canada
durocher@cs.mcgill.ca
[2] CNRS & LIRMM, Montpellier, France
paul@lirmm.fr

Abstract. Let C denote a set of n mobile clients, each of which follows a continuous trajectory on a weighted tree T. We establish tight bounds on the maximum relative velocity of the 1-centre and 2-centre of C. When each client in C moves with linear motion along a path on T we derive a tight bound of $\Theta(n)$ on the complexity of the motion of the 1-centre and corresponding bounds of $O(n^2\alpha(n))$ and $\Omega(n^2)$ for a 2-centre, where $\alpha(n)$ denotes the inverse Ackermann function. We describe efficient algorithms for calculating the trajectories of the 1-centre and 2-centre of C: the 1-centre can be found in optimal time $O(n \log n)$ when the distance function between mobile clients is known or $O(n^2)$ when the function must be calculated, and a 2-centre can be found in time $O(n^2 \log n)$. These algorithms lend themselves to implementation within the framework of kinetic data structures, resulting in structures that are compact, efficient, responsive, and local.

1 Introduction

Motivation. Finding a set of k points that are central to a collection of data points drawn from a metric space is a fundamental problem of geometry and data analysis. Within the context of facility location, this problem is commonly known as the k-centre problem; given a set P of points (clients) in a metric space S, a k-centre of P is a set of k points (facilities) such that the maximum distance from any client to its nearest facility is minimized. Two common choices for S are a Minkowski distance (typically ℓ_1, ℓ_2, or ℓ_∞) in Euclidean space and graph distance on a weighted graph.

Recently, the k-centre problem has been explored under mobility. In one dimension, the mobile 1-centre problem reduces to maintaining the extrema of a set of mobile clients [1,2,4,16]. Natural generalizations of this problem to higher dimensions in $\mathbb{R}^d$ lead to the mobile Euclidean 1-centre [2,6,10], the mobile rectilinear 1-centre [2,6], and the kinetic convex hull [4,5,16]. Although some mobile k-centre problems can be modelled by motion in Euclidean space, several applications are better represented by motion on a graph. That is, the underlying

⋆ Research for this work was supported by NSERC.

T. Tokuyama (Ed.): ISAAC 2007, LNCS 4835, pp. 341–352, 2007.

graph remains fixed while clients and facilities move along its edges and vertices. Examples include vehicles moving along a road network or mobile robots following defined routes in an industrial setting [7].

Although the static k-centre problem on graphs is well understood, the corresponding mobile problem remained unexplored. Any path in a weighted graph is isometric to a line segment; we generalize the motion of a single client on the line to motion on a path in a graph. That is, given a weighted graph G, each mobile client follows a continuous trajectory along the edges and vertices of G. Continuity and bounded velocity are natural constraints on any physical moving object. It is straightforward to show that for any graph G that contains a cycle, there exist sets of mobile clients on G whose 1-centre is discontinuous. As such, we primarily focus our attention on metric spaces for which the k-centre is continuous. In particular, graph distance on a tree maintains many properties of Euclidean distance in $\mathbb{R}^d$, such as a unique shortest path between two points and a unique, continuous 1-centre, while introducing interesting algorithmic challenges to the problem of maintaining a mobile k-centre.

Main Results. The 1-centre on a tree is unique [18]. We show its motion is continuous and has relative velocity at most one. Since a 2-centre of a tree is not unique, we identify a particular 2-centre which we call the *equidistant 2-centre* and show that its motion is continuous and has relative velocity at most two. The 3-centre is discontinuous even on a line segment; furthermore, no bounded-velocity approximation is possible for the mobile 3-centre [9]. We consider values of k for which the mobile k-centre is continuous: $k \leq 2$.

When each client in C moves with linear motion along a path on T, the motions of the corresponding 1-centre and equidistant 2-centre are piecewise linear. We derive a tight bound of $\Theta(n)$ on the complexity of the motion of the 1-centre, an upper bound of $O(n^2\alpha(n))$ on the complexity of the motion of the equidistant 2-centre, and a worst-case lower bound of $\Omega(n^2)$ on the complexity of the motion of any 2-centre, where $\alpha(n)$ denotes the inverse Ackermann function. We describe efficient algorithms for calculating the trajectories of the 1-centre and 2-centre of C. When the all-pairs distance function between mobile clients is known at all times, the 1-centre can be found in optimal time $O(n \log n)$. The distance function can be calculated in time $O(n^2)$. The equidistant 2-centre can be found in time $O(n^2 \log n)$. Moreover, our algorithms have natural implementations as kinetic data structures (KDS), resulting in structures that are compact, efficient, responsive, and local. Although previous applications of KDSs have been to mobile problems in Euclidean space, as we demonstrate, the KDS framework lends itself naturally to mobile problems on graphs.

2 Definitions

Since a *point* refers to a fixed position in a metric space, we refer to a *client* in the context of motion. Let $C = \{c_1, \ldots, c_n\}$ denote a set of mobile clients, where $I = [0, t_f]$ denotes a time interval, U_T denotes the continuum of points defined by a weighted tree $T = (V, E)$, and each c_i is a continuous function $c_i : I \to U_T$. For every $t \in I$, let $C(t) = \{c(t) \mid c \in C\}$ denote the set of points in U_T that corresponds to the positions of clients in C at time t. The position of a mobile *facility* f is a function of the positions of a set of clients, $f : \mathscr{P}(U_T) \to U_T$, where $\mathscr{P}(A)$ denotes the power set of set A.

A common assumption in problems involving motion in Euclidean space is that the position of a mobile client is a linear function over time (e.g., [1,2,4]). We make a similar assumption and consider clients with linear motion on trees to establish combinatorial bounds. A mobile client or facility a has *linear motion* if for all $t \in I$, $d(a(0), a(t)) = t \cdot v_a$, where v_a is a non-negative constant and $d(b, c)$ denotes the graph distance between points b and c in U_T. We refer to v_a as the *velocity* of a. The union of the trajectories of a set of n mobile clients that move with linear motion is a subgraph of U_T that has at most $2n$ vertices of degree one. Therefore, we assume that T has at most $2n$ leaves and at most $4n - 1$ vertices, and that $c(0)$ and $c(t_f)$ are vertices of T, for each $c \in C$.

We assume an upper bound of one on the velocity of clients since we are interested in *relative velocity*. Unlike mobile clients, a mobile facility is not required to travel along a path in T nor is its velocity required to remain constant. A mobile facility f has *maximum velocity* v_f if

$$\forall t_1, t_2 \in I,\ d(f(C(t_1)), f(C(t_2))) \leq v_f |t_1 - t_2|, \tag{1}$$

for all sets of mobile clients C defined on any tree T and any time interval I. Continuity is a necessary condition for any fixed upper bound on velocity.

We say client $c \in C$ is *extreme* at time t if $c(t)$ does not lie in the interior of any path through T between two clients in $C(t)$. The *convex hull* of $C(t)$ corresponds to the union of all paths between two clients in $C(t)$.

Definition 1. *Given a weighted tree T and a set of points C in U_T, a k-*centre *of C is a set of k points in U_T, denoted $\Xi_1(C), \ldots, \Xi_k(C)$, that minimizes*

$$\max_{c \in C} \min_{1 \leq i \leq k} d(c, \Xi_i(C)). \tag{2}$$

When $k = 1$, we omit the subscript and write $\Xi(C)$. The definition of a mobile k-centre of a set of mobile clients C follows directly from this static definition.

We refer to (2) as the *k-radius* of C or simply as its *radius* when $k = 1$. The diameter of C is twice the radius of C [19] (for graphs, the diameter is at most twice the radius). A *diametric path* of C is a path between two clients c_1 and c_2 in C such that the distance between them is the diameter of C. We refer to $\{c_1, c_2\}$ as a *diametric pair* and to c_1 and c_2 as *diametric clients*. The 1-centre of C is the unique midpoint of all diametric paths of C [18].

The 1-centre problem on graphs is also known as the absolute centre [18,19], single centre [19], and minimax location problem [8,18]. A common variation of the k-centre problem on graphs is known as the vertex k-centre or discrete k-centre problem, for which the choice of locations for the facility is restricted to vertices (clients) of the graph G. Maintaining continuity in the motion of a mobile facility is impossible in the vertex centre model, as a facility could be required to jump discontinuously from vertex to vertex (client to client).

3 Related Work

Handler [18] gives linear-time algorithms for identifying the 1-centre and 2-centre of a tree. Frederickson gives a linear-time algorithm for finding a k-centre of a tree when k is fixed [13]. Kariv and Hakimi [23] provide an $O(mn+n^2 \log n)$-time algorithm for the 1-centre problem on graphs, where $n = |V|$ and $m = |E|$. See [12,17,20,23,24,25] for reviews of k-centre problems on trees and on graphs.

Kinetic data structures (KDS), introduced by Basch et al. [4], allow the maintenance of an attribute (called the configuration function) of a set of mobile objects moving continuously in some metric space. To do so, a KDS maintains a dynamic set of certificates that guarantees the correctness of the configuration function at any time during the motion. Each certificate c is associated with a small set of mobile objects for which some property is verified. The failure time of certificate c (called an event) is calculated as a function of the motion of these objects. The failure time is added to a priority queue. Restoring the configuration function following a certificate failure requires updating the set of certificates (and the corresponding events in the queue).

Guibas [16] describes four properties used to evaluate the quality of a KDS. A KDS is *compact* if the maximum number of certificates active at any given time is linear in the degrees of freedom of the set of moving objects. A KDS is *responsive* if the maximum number of certificates associated with any one mobile object is polylogarithmic in the problem size. A KDS is *local* if at most a small number of certificates require updating as a result of a certificate failure. A KDS is *efficient* if the total number of certificate failures is proportional to the number of external events (changes to the configuration function). See [3,4,5,16] for a more complete description of the KDS framework.

In relation to our work on the mobile k-centre, KDSs have been constructed to maintain various attributes of a set of mobile clients; these include extremal elements in $\mathbb{R}$ [1,2,4,16], extent (e.g., diameter and width) in $\mathbb{R}^2$ [1,2], approximations of the mobile 1-centre in $\mathbb{R}^2$ [2,6,9,10], approximations of mobile 2-centres in $\mathbb{R}^2$ [11], the kinetic convex hull [4,5,16], an approximation of mobile k-centres in $\mathbb{R}^d$ [15], and approximations of discrete rectilinear k-centres [14,22].

4 The Mobile 1-Centre on Trees

4.1 Properties of the Mobile 1-Centre

The mobile 1-centre is continuous in $\mathbb{R}^d$ [9]. Although the mobile 1-centre has at most unit relative velocity in $\mathbb{R}$, its relative velocity is unbounded in $\mathbb{R}^2$ [6]. It can be shown that the mobile 1-centre is discontinuous on graphs. Restricted to trees, however, we show that the mobile 1-centre remains continuous and has at most unit relative velocity.

Theorem 1. *The mobile 1-centre has relative velocity at most one on trees. This bound is tight.*

Proof. Choose any $t_1, t_2 \in I$ and let $\delta = |t_1 - t_2|$. If $\Xi(C(t_1)) = \Xi(C(t_2))$, then (1) holds trivially. Therefore, assume $\Xi(C(t_1)) \neq \Xi(C(t_2))$. Let P denote the path in T between $\Xi(C(t_1))$ and $\Xi(C(t_2))$. Let r_1 and r_2 denote the respective radii of $C(t_1)$ and $C(t_2)$. Let L_1 denote the subtree of T that includes all branches of $\Xi(C(t_1))$ except P. Note, L_1 includes $\Xi(C(t_1))$. Similarly, let L_2 denote the subtree of T that includes all branches of $\Xi(C(t_2))$ except P.

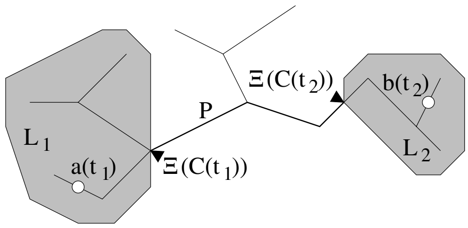

Fig. 1. illustration in support of Theorem 1

Let a be a client in C such that $a(t_1) \in L_1$ and $d(a(t_1), \Xi(C(t_1))) = r_1$. Similarly, let b be a client in C such that $b(t_2) \in L_2$ and $d(b(t_2), \Xi(C(t_2))) = r_2$. Such clients must exist since $\Xi(C(t))$ is the midpoint of a diametric path of $C(t)$ for all t. See Fig. 1. Therefore,

$$\begin{aligned} d(a(t_1), b(t_2)) &\leq d(a(t_1), \Xi(C(t_1))) + d(\Xi(C(t_1)), b(t_1)) + d(b(t_1), b(t_2)) \\ &\leq 2r_1 + \delta, \qquad \text{(3a)} \\ \text{and } d(a(t_1), b(t_2)) &\leq d(a(t_1), a(t_2)) + d(a(t_2), \Xi(C(t_2))) + d(\Xi(C(t_2)), b(t_2)) \\ &\leq 2r_2 + \delta. \qquad \text{(3b)} \end{aligned}$$

Consequently,

$$\begin{aligned} d(a(t_1), b(t_2)) &= d(a(t_1), \Xi(C(t_1))) + d(\Xi(C(t_1)), \Xi(C(t_2))) + d(\Xi(C(t_2)), b(t_2)), \\ \Rightarrow d(\Xi(C(t_1)), \Xi(C(t_2))) &= d(a(t_1), b(t_2)) - d(a(t_1), \Xi(C(t_1))) - d(\Xi(C(t_2)), b(t_2)) \\ &= d(a(t_1), b(t_2)) - r_1 - r_2 \\ &\leq \delta, \end{aligned}$$

by (3a) and (3b). The bound is realized when the two diametric clients move in a parallel direction. □

It follows that the mobile 1-centre is continuous on trees.

4.2 Complexity of the Motion of the 1-Centre

When n clients move along the real line, each with some constant velocity, the identity of the client that realizes either extremum changes $\Theta(n)$ times in the worst case [4]. In particular, any given client realizes each extremum at most once in the sequence of changes. When n clients move in $\mathbb{R}^2$ along linear trajectories with constant velocity, the diametric pair of clients changes $\Omega(n^2)$ times in the worst case [1]. As we show in Theorem 2, for a set C of n clients with linear motion on a tree T, the identity of the diametric pair of C changes $\Theta(n)$ times in the worst case. We begin with a definition.

Definition 2. *Given a client c moving with velocity v_c, the* outward velocity *of c at time t, denoted $\vec{v}(c(t))$, is given by*

$$\vec{v}(c(t)) = \begin{cases} -\infty \text{ if } c(t) \text{ is not extreme in } C(t), \\ -v_c \text{ if } c(t) \text{ moves towards the interior of the convex hull of } C(t), \\ \;\; v_c \text{ otherwise.} \end{cases}$$

Lemmas 1 through 3 assume linear motion of a set of clients C on a tree T. In addition, we assume that the diameter of C is non-zero at all times; a zero diameter implies that all clients in C coincide in a point and any two clients define a diametric pair. Furthermore, the interior of the convex hull is empty and, consequently, outward velocity is ill defined. We consider a zero diameter in the proof of Theorem 2.

Lemma 1. *The outward velocity of client $c \in C$ is non-decreasing while c remains in a diametric pair of C.*

Proof. Two cases are possible while c remains in a diametric pair of C.

Case 1. Assume c moves away from the interior of the convex hull of C initially. Client c has linear motion along a path $P \subseteq T$. The subpath of P that remains to be travelled by c lies outside the convex hull of C. Therefore the outward velocity of c remains constant.

Case 2. Assume c moves towards the interior of the convex hull of C initially. The outward velocity of c remains constant until c branches and turns away from the interior of the convex hull. The remainder of the motion corresponds to Case 1. □

As we show in Lemma 2, any change in the outward velocity at either endpoint of a diametric path must be increasing.

Lemma 2. *Choose any $t_1 \in I$ and let $\{a_1, b_1\}$ be a diametric pair of $C(t_1)$. If $\{a_2, b_2\}$ is a diametric pair of $C(t_2)$ and a_1 is not in any diametric pair of $C(t_2)$ for some $\epsilon > 0$ and all $t_2 \in (t_1, t_1+\epsilon)$, then $\vec{v}(a_1(t_1)) < \min\{\vec{v}(a_2(t_2)), \vec{v}(b_2(t_2))\}$.*

Proof. Since a_1 is in a diametric pair of $C(t_1)$,

$$\forall c \in C,\; d(a_1(t_1), b_1(t_1)) \geq d(a_1(t_1), c(t_1)). \tag{4}$$

Since $\{a_2, b_2\}$ is a diametric pair of $C(t_2)$ but a_1 is not in any diametric pair of $C(t_2)$,

$$\forall c \in C,\; d(a_1(t_2), c(t_2)) < d(a_2(t_2), b_2(t_2)). \tag{5}$$

Since client motion is continuous and by (4) and (5),

$$d(a_1(t_1), b_1(t_1)) = d(a_2(t_1), b_2(t_1)). \tag{6}$$

The result follows from (5) and (6). □

Lemma 3. *A client $c \in C$ becomes an endpoint of a diametric path of C at most four times.*

Proof. By Definition 2, the outward velocity of a client c in a diametric pair (c is extreme) is one of two values: $\pm v_c$. By Lemma 2, a change in a diametric pair corresponds to an increase in outward velocity. Therefore, a client c realizes either endpoint of a diametric path at most twice, for a total of at most four times. □

Theorem 2. *When each client in C moves with linear motion along a path on T, the motion of the 1-centre of C is piecewise linear and is composed of $\Theta(n)$ linear segments in the worst case, where $n = |C|$.*

Proof. Case 1. Assume the diameter of C is non-zero throughout the motion. The upper bound $O(n)$ follows from Lemma 1 and 3 and the fact that the 1-centre of C is the midpoint of a diametric pair.

Case 2. Assume the diameter of C is zero at some time during the motion. A zero diameter implies that all clients in C coincide at a point; that is, all clients cross simultaneously. This degeneracy occurs at most once since any two clients cross at most once. Since clients in C have linear motion, the motion of the 1-centre of C has linear motion while all clients coincide. Before and after the degeneracy, the motion of clients in C corresponds to Case 1. Therefore, the sum of the number of linear segments of the motion of the 1-centre remains $O(n)$.

The worst-case lower bound of $\Omega(n)$ follows from the corresponding result in one dimension [4]. □

4.3 Kinetic Maintenance of the Mobile 1-Centre

Given a set C of n mobile clients, each moving with linear motion in $\mathbb{R}$, the 1-centre of C is the midpoint of the extrema of C. The position of each extremum is given by the upper (respectively, lower) envelope of the set of n linear functions that correspond to the positions of clients in C relative to a fixed point in $\mathbb{R}$. Hershberger [21] gives an $O(n \log n)$ time algorithm which finds the upper envelope by dividing the set of linear functions in two, recursively finding the upper envelope of each set, and recombining the two envelopes to give the global upper envelope.

Using a related idea, we describe an algorithm for identifying a sequence of diametric pairs of a set of mobile clients, each moving with linear motion on a tree. We then describe how to implement the algorithm as a KDS. The algorithm makes use of the distance function d, where $d(a(t), b(t))$ returns the graph distance on tree T between mobile clients a and b at time t. We begin with the following lemma upon which our algorithm relies.

Lemma 4. *Let C_1 and C_2 be sets of points on U_T for some tree T. Let $\{a_i, b_i\}$ denote a diametric pair of C_i, for $i = 1, 2$. Set $\{e, f\}$ is a diametric pair of $C_1 \cup C_2$, where*

$$\{e, f\} = \underset{\{e', f'\} \subseteq \{a_1, b_1, a_2, b_2\}}{\operatorname{argmax}} d(e', f'). \tag{7}$$

The proof of Lemma 4 was omitted due to space limitations.

Algorithm Description. The set of mobile clients C is partitioned arbitrarily into sets C_1 and C_2 of size $\lfloor n/2 \rfloor$ and $\lceil n/2 \rceil$. For each $i = 1, 2$, the algorithm is called recursively to find a sequence of diametric pairs of C_i, denoted $\{a_{i,1}, b_{i,1}\}, \ldots, \{a_{i,m_i}, b_{i,m_i}\}$, and a corresponding partition of the time interval I, denoted $I_{i,1}, \ldots, I_{i,m_i}$, such that for each j, $a_{i,j}(t)$ and $b_{i,j}(t)$ are a diametric pair of $C_i(t)$ for all $t \in I_{i,j}$. The recursion terminates when $n \leq 2$, in which case each client in C is in a diametric pair. We now describe how to compute a corresponding sequence for C.

Consider a third partition of the time interval I, denoted $I_1, \ldots, I_m$, such that for each i, $I_i = I_{1,j} \cap I_{2,k}$, for some j, k. For all $t \in I_i$, diametric pairs of $C_1(t)$ and $C_2(t)$ consist of four clients in C, say a_1, b_1, a_2, and b_2. Let e and f be defined as in (7). By Lemma 4, e and f are a diametric pair of $C(t)$. The sequence of pairs of clients in $\{a_1, b_1, a_2, b_2\}$ that realize e and f corresponds to the sequence of pairs whose relative distance is maximized. That is, there are six combinations of pairs in $\{a_1, b_1, a_2, b_2\}$, each of which corresponds to an inter-client distance function. The upper envelope of these six functions determines the sequence of identities of e and f during I_i. Thus, solutions to the recursive subproblems are combined to find the sequence of diametric pairs of C.

Time Complexity. By Theorem 2, the complexity of the motion of the 1-centres of C_1 and C_2 is $O(n)$. That is, the time interval I can be partitioned into $O(n)$ subintervals such that the motion of each 1-centre is linear within every subinterval (i.e., $m \in O(n)$). Within each subinterval, we find the maximum of six piecewise-linear functions, each composed of at most four linear segments. Therefore, the maximum function is also piecewise linear, consists of at most 24 linear segments, and can be found in constant time. Thus, the solutions to the two subproblems are combined in $O(n)$ time. The recursion tree has depth $\lfloor \log_2 n \rfloor$, resulting in a total runtime of $O(n \log n)$. The worst-case lower bound of $\Omega(n \log n)$ follows from the corresponding one-dimensional problem [21].

Distance Function. Depending on the formulation of the problem, the input may not include the distance function. In this case, the input is given simply as the set of clients, each of which specifies an origin and destination vertex pair in T. In particular, the path of a client's trajectory is not given.

We assume only a basic weighted edge adjacency list or matrix for the tree T. Build a table $A[i, j]$ that stores the following information for each vertex u_i and each client c_j: $d(u_i, c_j(0))$, the velocity of $c_j(0)$ relative to u_i, and the instant in I (if any) at which the velocity of c_j relative to u_i becomes negated (that is, c_j takes a branch such that its motion changes from towards u_i to away from u_i). This information encodes the two-segment piecewise-linear function $d(u_i, c_j(t))$. Table $A[i, j]$ has size $O(n^2)$ and can be calculated in time $O(n^2)$ by considering each client c_j and tracing its trajectory through T.

For any clients c_1 and c_2 in C, the client-to-client distance function $d(c_1(t), c_2(t))$ can be calculated in constant time from table A. While c_1 and c_2 move towards each other, $d(c_1(t), c_2(t)) = |d(c_1(0), c_1(t)) - d(c_1(0), c_2(t))|$. After one client, say c_1, turns away from the other, $d(c_1(t), c_2(t)) = |d(c_1(t_f), c_1(t)) - d(c_1(t_f), c_2(t))|$.

KDS Implementation. We describe a KDS that maintains a diametric pair over time along with a set of certificates that validates the identity of the pair at any time during the motion.

Theorem 3. *Given a tree T and a set of mobile clients C, each moving with linear motion on a path of T, there exists a KDS to maintain the mobile 1-centre of C that is local, responsive, efficient, and compact.*

Proof. The set of certificates corresponds to the recursive hierarchy described in our algorithm. At any time t, for each set C in the hierarchy, the certificate for $C(t)$ consists of five inequalities that confirm the maximum of six functions. That is, the certificate verifies the identity of a diametric pair of $C(t)$ in terms of the diametric pairs of the subsets $C_1(t)$ and $C_2(t)$ by Lemma 4. The corresponding properties are certified recursively for $C_1(t)$ and $C_2(t)$. Each set maintains a single certificate defined in terms of four clients and the total number of certificates is $O(n)$; therefore, the KDS is compact. Each client is contained in at most $O(\log n)$ sets and, consequently, is associated with at most $O(\log n)$ certificates. As a result, a motion plan update for a client results in changes to the failure times of $O(\log n)$ certificates; therefore, the KDS is local.

A certificate failure occurs whenever the diametric pair of a set C changes. Locally, the certificate for C is restored in constant time; however, a change in the diametric pair of C may percolate upwards in the tree, resulting in $O(\log n)$ additional certificate updates; therefore, the KDS is responsive. By Theorem 2, each set C contributes at most $O(|C|)$ certificate failures, resulting in a total of $O(n \log n)$ certificate failures over the entire motion. Although this value is asymptotically greater than $\Theta(n)$ (the worst-case number of external events for a set of n clients), any offline algorithm for finding the trajectory of the 1-centre requires $\Omega(n \log n)$ time in the worst case, even in one dimension [21]. Therefore, the KDS is efficient. □

5 The Mobile 2-Centre on Trees

5.1 Properties of the Mobile 2-Centre

Although a 2-centre of a set of clients C on a tree is not unique (this is the case even in one dimension [9]), any 2-centre of C, $\Xi_1(C)$ and $\Xi_2(C)$, defines a natural bipartition of C, denoted $\{C_1, C_2\}$, such that

$$\forall c \in C_1,\ d(c, \Xi_1(C)) \leq d(c, \Xi_2(C)) \text{ and } \forall c \in C_2,\ d(c, \Xi_1(C)) \geq d(c, \Xi_2(C)).$$

We refer to $\{C_1, C_2\}$ as a *diametric partition* of C. A diametric partition induced by a given 2-centre is not unique. We refer to the *local 1-centre*, *local radius*, and *local diametric pair/path*, respectively, in reference to the 1-centre, radius, and diametric pair/path of C_1 or C_2. The local 1-centres of C_1 and C_2 are a 2-centre of C [19]. Proofs of results in Sect. 5 were omitted due to space limitations.

5.2 Equidistant 2-Centre

Even in one dimension the motion of a 2-centre defined by two local 1-centres is not continuous. This is easily demonstrated by an example: position a client at each endpoint of a line segment and let a third client move from one endpoint to the other. Not all 2-centres are discontinuous; we describe a strategy for defining the positions of a 2-centre on a tree whose motion is continuous and whose relative velocity is at most two. We refer to this particular 2-centre as the *equidistant 2-centre*:

Definition 3. *Let* $\{a, b\}$ *be a diametric pair of* C*. An* equidistant 2-centre *of* C*, denoted* $\{\dot{\Xi}_1(C), \dot{\Xi}_2(C)\}$*, is a pair of points that lie on the path between* a *and* b *at a distance* ρ *from* a *and* b*, respectively, where* ρ *denotes the 2-radius of* C*.*

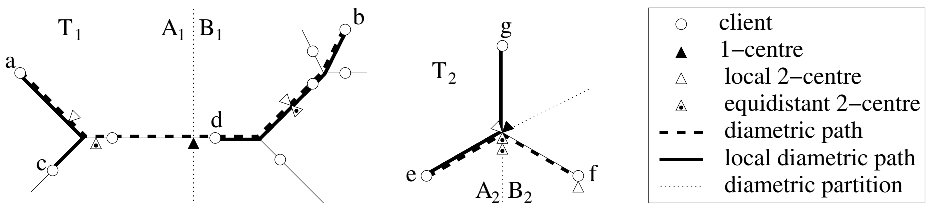

Fig. 2. Equidistant 2-centre examples

See Fig. 2 for an example. It is not difficult to show that the equidistance 2-centre of C is unique and that it is a 2-centre of C. It follows that the equidistant 2-centre is independent of the choice of the diametric pair $\{a, b\}$.

Theorem 4. *Each facility in the mobile equidistant 2-centre has relative velocity at most two.*

It follows that each facility in the mobile equidistant 2-centre is continuous. Since no mobile 2-centre can guarantee relative velocity less than two in one dimension [9], the maximum velocity of the equidistant 2-centre is optimal.

5.3 Complexity of the Motion of the 2-Centre

We establish the following bounds on the complexity of the motion of 2-centres:

Theorem 5. *When each client in* C *moves with linear motion along a path on* T*, the motion of each facility in the equidistant 2-centre of* C *is piecewise linear and is composed of* $O(n^2\alpha(n))$ *linear segments, where* $n = |C|$*.*

Theorem 6. *There exists a set of mobile clients* C*, each moving with linear motion in* $\mathbb{R}$*, such that the motion of some facility in any 2-centre of* C *whose motion is piecewise linear is composed of* $\Omega(n^2)$ *linear segments, where* $n = |C|$*.*

5.4 Kinetic Maintenance of the Mobile 2-Centre

Capitalizing on our 1-centre results, we describe an algorithm for identifying local 1-centres and the equidistant 2-centre of a set of mobile clients.

Algorithm Description. We first run our 1-centre algorithm to find a sequence of diametric pairs of C, denoted $\{a_1, b_1\}, \ldots, \{a_m, b_m\}$, and a corresponding partition of the time interval I, denoted $I_1, \ldots, I_m$, such that $m \in O(n)$. For each time interval I_i, determine when each client c is closer to a_i and when it is closer to b_i. This determines the sets $C_1(t)$ and $C_2(t)$ for all $t \in I_i$. Consider C_1 (an analogous algorithm applies to C_2). A diametric pair of $C_1(t)$ is given by $a_i(t)$ and a furthest client from $a_i(t)$ in $C_1(t)$. Each local diametric pair determines the motion of the corresponding local 1-centre and the local radius, from which the motion of the equidistant 2-centre is straightforward to calculate.

Time Complexity. For a client $c \in C$, the functions $d(c(t), a_i(t))$ and $d(c(t), b_i(t))$ are piecewise linear, each composed of at most four linear segments. Therefore, c changes partitions $O(1)$ times during interval I_i and calculating the interval for which c resides in either partition is achieved in constant time. Finding a furthest client from $a_i(t)$ for all $t \in I_i$ corresponds to finding the upper envelope of $n - 2$ partially-defined, piecewise-linear functions, which can be done in $O(n \log n)$ time using Hershberger's [21] algorithm. Since there are $O(n)$ time intervals, the total runtime is $O(n^2 \log n)$.

Theorem 7. *Given a tree T and a set of mobile clients C, each moving with linear motion on a path of T, there exists a KDS to maintain the mobile equidistant 2-centre of C that is compact and has responsiveness $O(n)$, locality $O(n)$, and efficiency $O(n^2 \log n)$.*

6 Directions for Future Research

As mentioned in Sect. 1, the mobile 1-centre is discontinuous on any cyclic graph. This motivates the search for bounded-velocity approximations of the k-centre on graphs. For the 1-centre, we have preliminary results showing that no continuous $(2 - \epsilon)$-approximation is possible for any $\epsilon > 0$. A unit-velocity 2-approximation is given by selecting an arbitrary client $c \in C$ and setting the position of the facility to coincide with $c(t)$. It is unknown whether any bounded-velocity approximation exists for mobile 2-centres on graphs. Finally, it may be possible to extend this work to maintain a discrete 1-centre and 2-centre of C.

Acknowledgements. The authors would like to thank David Kirkpatrick for suggesting that mobile k-centres might be interesting to consider on trees.

References

1. Agarwal, P.K., Guibas, L.J., Hershberger, J., Veach, E.: Maintaining the extent of a moving point set. Disc. & Comp. Geom. 26, 353–374 (2001)
2. Agarwal, P.K., Har-Peled, S.: Maintaining approximate extent measures of moving points. In: SODA, pp. 148–157 (2001)

3. Basch, J.: Kinetic Data Structures. PhD thesis, Stanford U. (1999)
4. Basch, J., Guibas, L., Hershberger, J.: Data structures for mobile data. J. Alg. 31, 1–28 (1999)
5. Basch, J., Guibas, L.J., Silverstein, C., Zhang, L.: A practical evaluation of kinetic data structures. In: SOCG, pp. 388–390 (1997)
6. Bespamyatnikh, S., Bhattacharya, B., Kirkpatrick, D., Segal, M.: Mobile facility location. In: Int. ACM Work. on Disc. Alg. & Meth. for Mob. Comp. & Comm, pp. 46–53 (2000)
7. Bruno, G., Ghiani, G., Improta, G.: Dynamic positioning of idle automated guided vehicles. J. Int. Man. 11, 209–215 (2000)
8. Dearing, P.M., Francis, R.L.: A minimax location problem on a network. Trans. Sci. 8, 333–343 (1974)
9. Durocher, S.: Geometric Facility Location under Continuous Motion. PhD thesis, U. of British Columbia (2006)
10. Durocher, S., Kirkpatrick, D.: The Steiner centre: Stability, eccentricity, and applications to mobile facility location. Int. J. Comp. Geom. & App. 16, 345–371 (2006)
11. Durocher, S., Kirkpatrick, D.: Bounded-velocity approximations of mobile Euclidean 2-centres. Int. J. Comp. Geom. & App. (to appear, 2007)
12. Dvir, D., Handler, G.Y.: The absolute center of a network. Networks 43, 109–118 (2004)
13. Frederickson, G.N.: Parametric search and locating supply centers in trees. In: Dehne, F., Sack, J.-R., Santoro, N. (eds.) WADS 1991. LNCS, vol. 519, pp. 299–319. Springer, Heidelberg (1991)
14. Gao, J., Guibas, L.J., Hershberger, J., Zhang, L., Zhu, A.: Discrete mobile centers. Disc. & Comp. Geom. 30, 45–65 (2003)
15. Gao, J., Guibas, L.J., Nguyen, A.: Deformable spanners and applications. Comp. Geom.: Th. & App. 35, 2–19 (2006)
16. Guibas, L.J.: Kinetic data structures: a state of the art report. In: Work. Alg. Found. Rob, pp. 191–209. A. K. Peters, Ltd (1998)
17. Halpern, J., Maimon, O.: Algorithms for the m-center problems: A survey. Eur. J. Oper. Res. 10, 90–99 (1982)
18. Handler, G.Y.: Minimax location of a facility in an undirected tree graph. Trans. Sci. 7, 287–293 (1973)
19. Handler, G.Y.: Finding two-centers of a tree: The continuous case. Trans. Sci. 12, 93–106 (1978)
20. Hansen, P., Labbé, M., Peeters, D., Thisse, J.F.: Single facility location on networks. An. Disc. Math. 31, 113–146 (1987)
21. Hershberger, J.: Finding the upper envelope of n line segments in $O(n \log n)$ time. Inf. Proc. Let. 33, 169–174 (1989)
22. Hershberger, J.: Smooth kinetic maintenance of clusters. Comp. Geom.: Th. & App. 31, 3–30 (2005)
23. Kariv, O., Hakimi, S.L.: An algorithmic approach to network location problems. I: The p-centers. SIAM J. App. Math. 37, 513–538 (1979)
24. Tansel, B.C., Francis, R.L., Lowe, T.J.: Location on networks: A survey. part I: The p-center and p-median problems. Man. Sci. 29, 482–497 (1983)
25. Tansel, B.C., Francis, R.L., Lowe, T.J.: Location on networks: A survey. part II: Exploiting tree network structure. Man. Sci. 29, 498–511 (1983)