Theoretical Computer Science 410 (2009) 5128–5137

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

(r, p)-centroid problems on paths and trees

J. Spoerhase*, H.-C. Wirth

Universität Würzburg, Lehrstuhl für Informatik I, Am Hubland, 97074 Würzburg, Germany

ARTICLE INFO

Article history:
Received 17 March 2009
Received in revised form 6 August 2009
Accepted 21 August 2009
Communicated by G. Ausiello

Keywords:
Competitive location
Computational complexity
Efficient algorithms
Trees

ABSTRACT

An instance of the (r, p)-centroid problem is given by an edge and node weighted graph. Two competitors, the leader and the follower, are allowed to place p and r facilities, respectively, into the graph. Users at the nodes connect to the closest facility. A solution of the (r, p)-centroid problem is a leader placement such that the maximum total weight of the users connecting to any follower placement is as small as possible.

We show that the absolute (r, p)-centroid problem is NP-hard even on a path which answers a long-standing open question of the complexity of the problem on trees (Hakimi, 1990 [10]). Moreover, we provide polynomial time algorithms for the discrete (r, p)-centroid on paths and the $(1, p)$-centroid on trees, and complementary hardness results for more complex graph classes.

1. Introduction

Location theory deals with problems of optimally placing facilities to serve the individual demands of a given set of users. Many problems in that area consider the case where one monopolistic provider places all of the facilities. In contrast, *competitive location* investigates scenarios where two (or more) providers place their facilities and users can decide between the providers. It is assumed that all facilities and all competitors provide the same type of good or service. Hence users decide on a facility serving their demand solely based on the distances to the servers. We assume a *binary demand rule*, i.e., for each user the total demand is served by exactly one facility.

In our scenario, the universe is modeled by a graph with weighted edges inducing distances. Weighted nodes of the graph represent users and their demand which is to be served by the competitors.

The benefit of each competitor is measured by the size of its *party*, i.e., the total demand (or weight) of the users connecting to the competitor. The providers act in a non-cooperative way and only aim at maximizing their own benefit. In the competitive location scenario investigated here, two competitors, called *leader* and *follower*, sequentially place p and r servers, respectively. Once the leader has chosen his facilities, say X_p, the follower is able to determine an optimal set of locations maximizing his benefit; such a solution is called an (r, X_p)-*medianoid*. Hence the follower's reaction is predictable, which the leader can take into account when he makes the initial decision, namely determining an (r, p)-*centroid*.

Similar questions arise in *voting location* problems on graphs [5]. Here, a set of users is asked to decide between two candidates by means of an election, while the user preference is determined by graph distances. Interesting solutions are particularly *stable* candidates, i.e., where there is no strong party of users who agree in preferring the same opposition over that candidate. In particular, a *p-Simpson* solution is a p-element candidate placement minimizing the influence of any possible p-element opposition; this can be equivalently formulated as a (p, p)-centroid problem.

Our paper is organized as follows. In Section 2 we investigate the complexity of the absolute and discrete (r, p)-centroid problem on paths and trees. These results answer the long-standing open question whether that problem is polynomial time solvable on trees [10]. Section 3 is devoted to algorithms for the absolute and discrete $(1, p)$-centroid on trees.

* Corresponding author. Tel.: +49 931 888 49 43.
E-mail addresses: joachim.spoerhase@uni-wuerzburg.de (J. Spoerhase), wirth@informatik.uni-wuerzburg.de (H.-C. Wirth).

0304-3975/$ – see front matter
doi:10.1016/j.tcs.2009.08.020

1.1. Problem definition

Consider an undirected graph $G = (V, E)$ with positive edge lengths $d\colon E \to \mathbb{Q}^+$. An edge of the graph can be considered as an infinite set of *points*. A point x on edge $e = (u, v)$ is specified by the distance from one of the endpoints of e, and the remaining distance is derived from the invariant $d(u, x) + d(x, v) = d(e)$. Notice that the set of points of a graph includes the set of nodes. All points which are not nodes are called *inner points*. In what follows we will use G (and e) both for denoting the graph (the edge) and for denoting all of its points, as the meaning will become clear from the context. In the sense of these considerations the edge length function d is extended to a distance function $d\colon G \times G \to \mathbb{Q}_0^+$ defined on all pairs of points. Nonnegative node weights $w\colon V \to \mathbb{Q}_0^+$ specify the demand of users who are always placed at nodes of the graph. Where appropriate we can assume w.l.o.g. that the edge lengths and node weights are integer numbers.

Let $X, Y \subset G$ be finite sets of nodes or points, specifying a server placement of the leader or follower player, respectively. The distance of a user u to a finite point set M is given by $d(u, M) := \min_{m \in M} d(u, m)$. A user u prefers the follower if $d(u, Y) < d(u, X)$. By $w(Y \prec X) := \sum\{\, w(u) \mid u \in V \text{ where } d(u, Y) < d(u, X)\,\}$ the total weight of the follower party is denoted. For convenience we write $w(y \prec X)$ instead of $w(\{y\} \prec X)$ for single element sets.

Let $r, p \in \mathbb{N}$ and $X_p \subset G$ be a set of $|X_p| = p$ points. Let

$$w_r^*(X_p) := \max_{\substack{Y_r \subset G \\ |Y_r| = r}} w(Y_r \prec X_p)$$

be the maximum influence any r-element follower placement can gain over the fixed leader placement X_p. An *absolute* (r, X_p)-*medianoid* of the graph is any set $Y_r \subset G$ of $|Y_r| = r$ points where $w(Y_r \prec X_p) = w_r^*(X_p)$ is attained. Let

$$w_{r,p}^* := \min_{\substack{X_p \subset G \\ |X_p| = p}} w_r^*(X_p).$$

An *absolute* (r, p)-*centroid* of the graph is any set $X_p \subset G$ of $|X_p| = p$ points where $w_r^*(X_p) = w_{r,p}^*$ is attained. The notions *discrete* (r, X_p)-*medianoid* and *discrete* (r, p)-*centroid* are defined similarly, with the server sets restricted to nodes, i.e., $X_p, Y_r \subseteq V$, rather than points.

Previous results and contribution of this paper

The (r, p)-centroid and (r, X_p)-medianoid problems have been introduced in [9]. Both of them are intractable on general graphs. The (r, p)-centroid problem is even Σ_2^{P}-complete on general graphs [15] while the $(1, p)$-centroid and (r, X_p)-medianoid are NP-hard [9]. The (r, X_p)-medianoid is solvable in time $O(rn^2)$ on a tree and in time $O(n)$ on a path [14]. The $(1, 1)$-centroid on a tree is equivalent to the 1-median [10] which can be determined in linear time [8]; on a general graph the $(1, 1)$-centroid can be found in polynomial time [11,4]. All of the above reported results apply to both absolute and discrete cases.

While the (r, X_p)-medianoid problem is efficiently solvable on trees, for many years the complexity status of the absolute (r, p)-centroid problem on trees was an open question [10,6,2]; see also [17] for a more recent overview. In this paper we prove that this problem is NP-hard even on paths. In contrast to that we show that the discrete (r, p)-centroid on a path can be solved in polynomial time, but becomes NP-hard on a spider. Finally, we give a polynomial time algorithm for discrete and absolute $(1, p)$-centroids on a tree and contrast this by showing NP-hardness for the same problem on pathwidth bounded graphs. To the best of our knowledge these are the first nontrivial results on certain graph classes where the (r, p)-centroid problem is polynomial time solvable.

In the model we are investigating, each customer attaches to exactly one server, and the weight of the user is constant and in particular does not depend on the distance to the selected server. This is known as an *inelastic binary* demand rule; see e.g. [17] for a review of other user demand rules.

Preliminaries

In our hardness proofs we make use of a reduction from the well-known PARTITION problem (problem SP12 in [7]):

Theorem 1.1 (*Hardness of* PARTITION). *The decision problem "Given a multiset $S = \{s_1, \ldots, s_n\}$ of integers with total sum $S^* := \sum S$, is there a sub-multiset $S' \subset S$ such that $\sum S' = \frac{1}{2} S^*$?" is NP-complete.* □

2. The (r, p)-centroid

In this section we investigate the complexity of the (r, p)-centroid problem where r, p are arbitrary integers specified as part of the input instance. The positive result is that the discrete (r, p)-centroid on a path can be computed efficiently. On the negative side, the same problem becomes NP-hard on slightly more complicated graphs, namely spiders. Moreover, the absolute (r, p)-centroid is already NP-hard on a path.

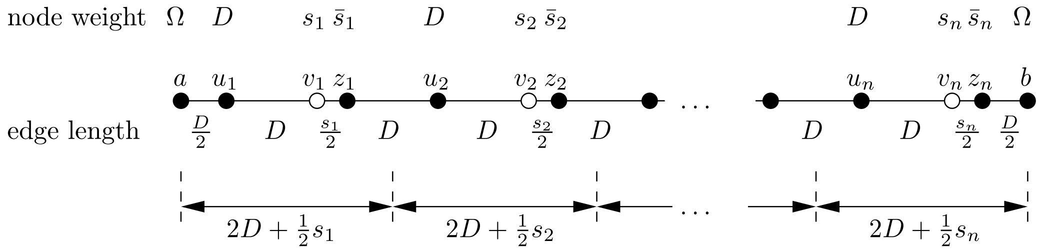

Fig. 1. Illustration of the path construction.

2.1. Absolute (r, p)-centroid on a path

In this section we show that the absolute (r, p)-centroid problem is already NP-hard when the underlying graph forms a path. To this end, let the path graph $G = (V, E)$ be given by its node set $V = \{v_1, \ldots, v_n\}$ and edge set $E = \{(v_1, v_2), \ldots, (v_{n-1}, v_n)\}$. Consider a leader placement $X_p = \{x_1, \ldots, x_p\} \subset G$ of $|X_p| = p$ points sorted such that $d(v_1, x_1) < \ldots < d(v_1, x_p)$. This defines a segmentation of the path into at most $p + 1$ disjoint intervals $T_0 := [v_1, x_1]$, $T_i := [x_i, x_{i+1}]$ for $i = 1, \ldots, p-1$, and $T_p := [x_p, v_n]$. By placing one server into an interval $[x_i, x_{i+1}]$, the follower can gain all nodes of any open interval $]a, b[\subset [x_i, x_{i+1}]$ of size $d(a, b) = d(x_i, x_{i+1})/2$. The best of these placements of the follower can be found with a simple linear time sweep algorithm.

Theorem 2.1 (*Absolute (r, p)-Centroid on Path*). *The absolute (r, p)-centroid problem is NP-hard on a path.*

Proof. Let an instance of problem Partition be given as in Theorem 1.1. Construct a path $P = (a, u_1, v_1, z_1, \ldots, u_n, v_n, z_n, b)$ with $3n + 2$ nodes (confer Fig. 1). To define the weights let $s_{\max} := \max_i s_i$ and $D := 2ns_{\max} + 1$ and $\Omega := 2nD + 1$. Let $w(a) := w(b) := \Omega$, and for all $i = 1, \ldots, n$ set $w(u_i) := D$, $w(v_i) := s_i$ and $w(z_i) := \bar{s}_i := D - s_i$. The nodes u_i, z_i are referred to as *heavy*, while the v_i nodes are called *light* nodes.

We define the edge lengths as follows: $d(a, u_1) := \frac{1}{2}D$, $d(u_i, v_i) := D$, $d(v_i, z_i) := \frac{1}{2}s_i$, $d(z_i, u_{i+1}) := D$, and $d(z_n, b) := \frac{1}{2}D$. The total length of the path is $2nD + \frac{1}{2}S^*$.

Set the number of leader positions to $p := n + 1$ and the number of follower positions to $r := n$. We will show in what follows that there is an (r, p)-centroid of gain $w^*_{r,p} \le n \cdot D + \frac{1}{2}S^*$ if and only if the instance of Partition admits a subset S' of sum $\frac{1}{2}S^*$.

"If": Assume that the instance of Partition is solvable with solution S', i.e., $\sum S' = \frac{1}{2}S^*$. Place two servers of the leader at the border nodes a, b. The remaining $n - 1$ leader servers divide the path into n intervals T_i of length t_i $(i = 1, \ldots, n)$. The interval division is called *valid* if for each $i = 1, \ldots, n$ the interval T_i contains the three nodes u_i, v_i, z_i as inner nodes. Choose the server positions such that $t_i := 2D + s_i$ if $s_i \in S'$ ("long interval") and $t_i := 2D$ otherwise ("short interval"). Observe that this yields a valid interval division. The gain of the follower in interval T_i when placing one server is D if it is a short interval and $D + s_i$ if it is a long interval (throughout this discussion we assume that the follower chooses positions maximizing his gain). There is no advantage in placing two servers into the same interval, as the gain would be $2D$ in that case. Hence we can assume w.l.o.g. that the follower places exactly one server per interval and thus achieves the total gain $nD + \frac{1}{2}S^*$.

"Only if": Consider the case of a leader placement with follower gain $w^*_r \le n \cdot D + \frac{1}{2}S^*$. We claim: The leader chooses a valid interval division.

It is clear that the leader places two servers at the two nodes a, b of weight Ω. Let $(t_i)_i$ $(i = 1, \ldots, n)$ be the sequence of interval lengths of the leader's placement.

Assume for contradiction that the right endpoint of some interval T_i is at the node z_i or to the left of it. The remaining $n - i$ intervals to the right of interval T_i cover a path length of at least $d(z_i, b) > 2(n - i)D + \frac{1}{2}D$, so by averaging there must be one interval of length larger than

$$\left(2 + \frac{1}{2(n-i)}\right)D > \left(2 + \frac{1}{2n}\right)D > 2D + s_{\max}.$$

By construction of the path, any interval of length larger than $2D + s_{\max}$ contains at least two heavy nodes which are inner nodes and within maximum distance of $D + \frac{1}{2}s_{\max}$. Hence, in that particular interval, the follower can gain both heavy nodes by placing a single server. Let $H := \min_i w(z_i) = D - s_{\max}$ be the minimum weight of heavy nodes. Placing the remaining $n - 1$ servers at free heavy nodes, this yields a total gain of at least

$$2H + (n-1)H = nD + D - (n+1)s_{\max} > nD + \frac{1}{2}S^*$$

for the follower, contradicting the premise. By an analogous argument we can show that the left endpoint of interval T_i does not lie at u_i or to the right of it. This shows the claim.

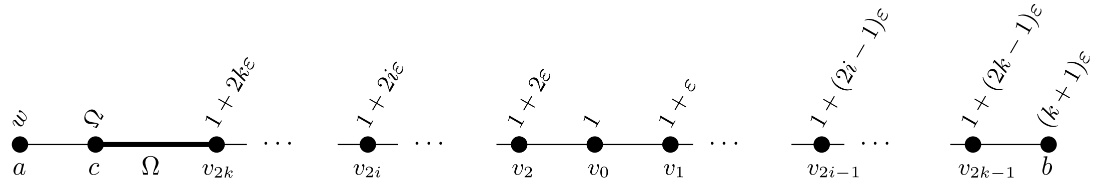

Fig. 2. The $(2, 2)$-centroid does not satisfy the optimal substructure property.

From this property we deduce that each interval left by the leader has inner nodes of total weight $2D$. Since the follower can always gain weight D by placing at u_i, we can assume w.l.o.g. that the follower places exactly one server into each interval. Moreover, the length of each interval T_i is bounded from above by $2D + s_i$: otherwise, the follower could cover all inner nodes of T_i with a single server, which would lead to a total gain of at least $2D + (n - 1)H > (n + 1)H$, contradicting the premise.

We distinguish two kinds of interval, namely those of length $t_i \leq 2D$, which we call *short intervals*, and those of length $2D < t_i \leq 2D + s_i$, called *long intervals*. We define the multiset $S' \subseteq S$ to be the multiset of those s_i where T_i is a long interval. As argued above, the follower places exactly one server into each interval T_i. This defines for each interval a number w_i denoting the follower's gain in that interval. Obviously $w_i = D$ for short intervals and $w_i = D + s_i$ for long intervals. This yields $t_i - D \leq w_i$. Hence

$$\frac{S^*}{2} = \sum_{i=1}^{n}(t_i - 2D) \leq \sum_{i=1}^{n}(w_i - D) \leq \frac{S^*}{2},$$

where the first equality follows from the path length $2nD + \frac{1}{2}S^*$ and the last inequality from the premise $w_r^* \leq nD + \frac{1}{2}S^*$. Thus we can conclude that $\sum S' = \sum_{i=1}^{n}(w_i - D) = \frac{1}{2}S^*$, which completes the proof. □

2.2. Discrete (r, p)-centroid on a path

Many optimization problems exhibit an *optimal substructure property* [3] (or *principle of optimality* [1]): essentially this means that a problem instance can be separated into independent subproblems such that optimal solutions of these subproblems can be combined to solve the original problem optimally. This property is exploited by widespread algorithmic techniques like divide and conquer, greedy, or dynamic programming.

In the case of the discrete (r, p)-centroid problem on a path this suggests the following hypothetical approach. Consider a path P with an (r, p)-centroid X_p and a node $x \in X_p$. Let P_1, P_2 be the subpaths resulting from splitting P at x. One could suspect that for suitable p_1, p_2, r_1, r_2 there are (r_i, p_i)-centroids on P_i such that their union forms an (r, p)-centroid on P, with the reasoning that no user in one subpath ever patronizes any server on the other subpath.

The example depicted in Fig. 2 shows that the (r, p)-centroid problem does not exhibit the optimal substructure property even when $r = p = 2$ and the underlying graph is a path. The path consists of $2k + 3$ nodes; the weights are given in the figure. Edges have unit length unless stated otherwise. Constant Ω is chosen large enough so that the leader always places one server at node c. An optimal substructure property would suggest that the weight w of the leftmost node a has no influence on the position of the second leader server in the subpath to the right of c.

However, this is not true. Let $w_r(i)$ $(r = 1, 2)$ be the maximum weight that the follower can claim when the leader places servers at c, v_i and the follower places r servers optimally on the node set $V - \{a\}$. It is easy to see that by picking i and setting $w := w_2(i) - w_1(i)$ we can enforce any of the nodes v_i to be the position of the second leader server.

As a consequence, a straightforward application of divide and conquer techniques cannot be successful in attacking the centroid problem on a path.

The algorithm

Let G be the input path with ordered vertex set $V = \{v_1, \ldots, v_n\}$. In order to compute a discrete (r, p)-centroid, we reduce this problem to the k-sum shortest path problem which was solved in [16] within a framework for general k-sum optimization problems where the underlying minisum problem is efficiently computable.

Definition 2.2 (*k-Sum Shortest Path*). A k-sum shortest s-t-path is a path from s to t where the sum of the k largest arcs is as small as possible.

We define a new digraph G' as depicted in Fig. 3. Start with a node set $V' := \{ v_{ij} \mid i = 1, \ldots, n \text{ and } j = 1, \ldots, p \}$. For any $i, j \in \{1, \ldots, n\}$, $i < j$, and any $k \in \{1, \ldots, p - 1\}$, add a path of two consecutive arcs (introducing a new vertex in the middle) from $v_{i,k}$ to $v_{j,k+1}$. This shall model the case that the leader places the kth server at v_i and the next server at v_j. Moreover, add new super nodes s, t to the graph and connect them by arcs from s to all v_{i1} and from all v_{ip} to t.

The lengths of the arcs are determined by the gain of the follower on partial intervals. Let $w_1(i, j)$ denote the maximum weight which a single follower server can claim on the partial interval between two leader servers placed at v_i and v_j. Similarly, let $w_2(i, j) = \sum_{\nu=i+1}^{j-1} w(v_\nu)$ be the maximum weight which can be claimed with two follower servers. (It is of no

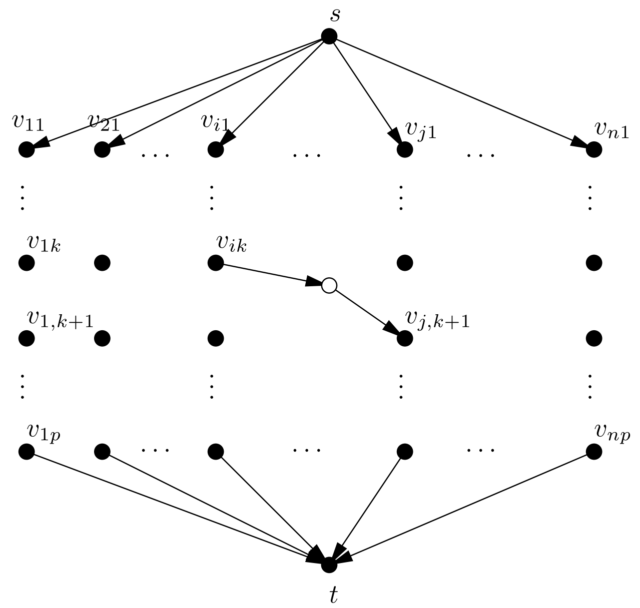

Fig. 3. Auxiliary graph to solve the discrete (r, p)-centroid on a path.

use placing more than two servers into a single interval.) For any path of two arcs connecting v_{ik} to $v_{j,k+1}$, set the length of the first arc to $w_1(i, j)$ and the length of the second arc to $w_2(i, j) - w_1(i, j)$. Finally, set the length of arcs (s, v_{i1}) to $\sum_{\nu=1}^{i-1} w(v_\nu)$ and that of arcs (v_{ip}, t) to $\sum_{\nu=i+1}^{n} w(v_\nu)$. This completes the construction of the acyclic graph G'.

Lemma 2.3. *The r-sum length of an s-t-path through nodes $v_{i_1,1}, \dots, v_{i_p,p}$ equals the weight of an (r, X_p)-medianoid where $X_p = \{v_{i_1}, \dots, v_{i_p}\}$.*

Proof. By construction, any s-t-path in G' meets exactly p nodes of the initial node set V'. This establishes a one to one relationship between placements of the p servers of the leader and s-t-paths in the auxiliary graph.

Observe that, for any $i < j$, $w_1(i, j) \le w_2(i, j) \le 2w_1(i, j)$. Therefore, the follower can achieve the maximum gain by a simple greedy strategy: Given the $p + 1$ intervals left by the leader, determine for each interval the gain w_1 of placing one server and the incremental gain $w_2 - w_1 \le w_1$ of placing two servers. The weight of the (r, X_p)-medianoid is the sum of the r largest numbers out of this multiset, which is also the r-sum length of the s-t-path in G'. □

The (r, p)-centroid minimizes the weight of (r, X_p)-medianoid over all server placements X_p, which corresponds to an r-sum minimization of paths in the graph G'. An r-sum shortest s-t-path in graph G' is equivalent to a solution of the (r, p)-centroid problem on path G.

Theorem 2.4 (*Discrete (r, p)-Centroid on Path*). *A discrete (r, p)-centroid of a path can be found in $O(pn^4)$.*

Proof. In [16] it has been shown that the k-sum optimization problem can be solved in $O(M \cdot t)$, where M is the number of different weights of items in the ground set and t is the time needed for solving one instance of the underlying minisum problem. In our setting, the set of ground elements is the set of arcs of size $O(pn^2)$ but with only $O(n^2)$ different weights. The minisum problem (shortest s-t-path in an acyclic graph of $O(pn^2)$ arcs) can be solved in time $O(pn^2)$. □

2.3. Discrete (r, p)-centroid on a tree

In this section we are going to show that determining a discrete (r, p)-centroid is NP-hard on a spider, i.e., a tree where exactly one node has degree larger than 2.

Theorem 2.5 (*Hardness of (r, p)-Centroid on a Spider*). *The problem of determining a discrete (r, p)-centroid on a spider is NP-hard.*

Proof. Let an instance of problem Partition be given as in Theorem 1.1. Construct a spider as depicted in Fig. 4. The node set consists of a central node c and for each integer s_i of a leg with nodes c–t_i–u_i–$\bar{u}_i$–v_i. The weight of the nodes is set to $w(c) := \Omega^3$, $w(t_i) := \Omega s_i$, $w(u_i) := \Omega^3$, $w(\bar{u}_i) := s_i$, and $w(v_i) := \Omega^3 + \Omega^2$. Finally, we add a special leg c–o–h–c' of weight $w(h) := \Omega^4$, $w(o) := 0$, and $w(c') := \Omega^3 + \frac{1}{2}\Omega S^*$. Here we choose $\Omega := 1 + nS^*$. All edges have unit length.

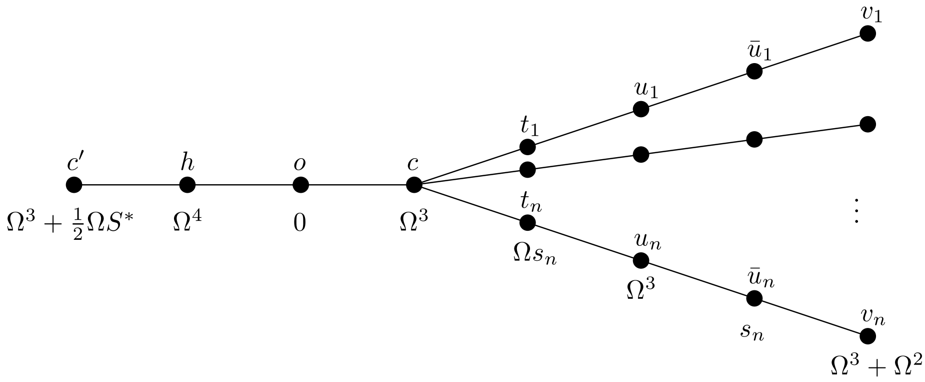

Fig. 4. The discrete (r, p)-centroid is NP-hard on a spider.

We set $r := p := n + 1$ and claim: There is a $(n + 1, n + 1)$-centroid of weight

$$W := (n + 1)\Omega^3 + n\Omega^2 + \frac{1}{2}S^*(\Omega + 1)$$

if and only if the instance of PARTITION is solvable.

"If": Let $S' \subseteq S$ with $\sum S' = \frac{1}{2}S^*$. Place a leader server at h; furthermore, for each i place another leader server at $\bar{u}_i$ if $s_i \in S'$ and at u_i otherwise. We look at the gain of the follower. Observe that it is not possible that the follower claims c and one of the u_i with a single server only. Since $w(c) + \sum_j w(t_j) < w(v_i)$ it is optimal to claim all peripheral nodes v_i. This is accomplished by placing a server at v_i if $s_i \in S'$ and at $\bar{u}_i$ otherwise. This way the follower claims all nodes v_i, $i = 1, \ldots, n$, and those nodes $\bar{u}_i$ where $s_i \notin S'$, with a weight of

$$n(\Omega^3 + \Omega^2) + S^* - \sum S' = n(\Omega^3 + \Omega^2) + \frac{1}{2}S^*.$$

The remaining server can be placed either at c' or at the central node c where it claims c and those nodes t_i where $s_i \in S'$. This contributes a weight of

$$\Omega^3 + \Omega \sum S' = \Omega^3 + \frac{1}{2}\Omega S^*,$$

which is the same for both cases. Adding both terms shows that the total weight of the (r, p)-centroid is exactly equal to W.

"Only if": In an optimal solution it is obvious that the leader places one server at the node h of weight Ω^4. Further, observe that there are enough nodes of weight Ω^3 or greater (namely the $2n+2$ nodes u_i, v_i, c, c') such that the follower can always place only at those nodes and thus gain at least Ω^3 per server.

We claim that the leader chooses on each leg either the node u_i or $\bar{u}_i$. If the leader places a server at central node c or at one of the t_i, then there are $n - 1$ additional servers left to place. This would leave at least one leg j free to the follower so that he could place at node u_j and gain both u_j and v_j of weight more than $2\Omega^3$ with a single server, resulting in a total of more than $(n + 2)\Omega^3$. As a consequence, the leader must choose, on each leg, either u_i, or $\bar{u}_i$, or v_i. If the leader were to place at the peripheral node v_i, then the follower could place at t_i, which would claim both u_i and the central node c with this server, which yields a similar contradiction. This shows the claim.

Let $S' := \{ s_i \mid \text{leader places server at } \bar{u}_i \} \subseteq S$ be the set of items where the leader places a server at the outer node in the corresponding leg. Suppose $\sum S' > \frac{1}{2}S^*$. Then the follower places on leg i next to the leader, claiming the nodes v_i, $i = 1, \ldots, n$, and the nodes $\bar{u}_i$ where $s_i \notin S'$. The remaining server is placed at the central node c and claims the nodes t_i where $s_i \in S'$. This yields a follower gain of

$$n(\Omega^3 + \Omega^2) + \left(S^* - \sum S'\right) + \Omega^3 + \Omega \sum S' > n(\Omega^3 + \Omega^2) + \Omega^3 + (\Omega + 1)\frac{1}{2}S^* = W,$$

where we make use of $\sum S' \geq \frac{1}{2}S^* + 1$ and $\Omega > S^*$. Suppose $\sum S' < \frac{1}{2}S^*$. Like above, the follower places n servers on the periphery; the remaining server is placed at c'. This yields a gain of

$$n(\Omega^3 + \Omega^2) + \left(S^* - \sum S'\right) + \Omega^3 + \frac{1}{2}\Omega S^* > W.$$

This completes the proof. □

3. The $(1, p)$-Centroid

We have pointed out in Section 2.2 that the (r, p)-centroid problem does not exhibit the optimal substructure property for $r \geq 2$. In this section we investigate the case $r = 1$ where this property holds.

3.1. Discrete $(1, p)$-centroid on a tree

First we consider the discrete $(1, p)$-centroid problem. Choose an arbitrary node $s \in V$, and connect s to a new node s_0 of weight 0 by an edge of length ∞. Then choose s_0 as the root of the tree. For any node $v \in V$ we denote by T_v the subtree hanging down from v. We can assume w.l.o.g. that the leader does not place at s_0 of zero weight.

Let $X \subseteq V - s_0$ be a node subset and $W \in \mathbb{N}$. Set X is called *W-bounding* if

1. $w_1^*(X) \leq W$ and
2. for all $x \in X$ with father x' we have $w_1^*(X - x + x') > W$.

Lemma 3.1. *If $w_{1,p}^* \leq W$ then $|X| \leq p$ for all W-bounding sets $X \subseteq V$.*

Proof. Assume that $w_{1,p}^* \leq W$ and let X^* with $|X^*| \leq p$ be an optimal leader placement. Consider an arbitrary W-bounding set X. Map each node from X^* to its closest ancestor in X (this allows us in particular to map a node to itself). We claim that this mapping is surjective, which completes the proof.

Assume for contradiction that there is a node $v \in X$ which is not in the image of the mapping, and let u be the father of v. By property 2 there is a $y \in T_u$ such that $w(y \prec X - v + u) > W$. Consider the maximal subtrees T' and T^* which contain the node y but no node from $X - v + u$ and X^*, respectively, as inner nodes. First, by its choice, y lies in the subtree T_u. Moreover, the closest ancestor of y in X is v (otherwise $w(y \prec X) = w(y \prec X - v + u) > W$). This implies that no inner node of T' can be part of X^*, for otherwise, v would be the image of this node, contradicting the premise. Hence T' is a subtree of T^*. Moreover, $w(y \prec X^*) \geq w(y \prec X - v + u) > W$, which is a contradiction. □

We propose the following algorithm: Initialize the node set X which shall be W-bounding at the end to $X \leftarrow \emptyset$. Start at the newly introduced root node s_0 and perform a depth first search traversal of the tree. Whenever the traversal returns from a node v back to its father u, perform the test whether there is an $y \in T_v$ such that $w(y \prec X + u) > W$. If this is the case, then add the node $X \leftarrow X + v$.

Lemma 3.2. *Given $W \in \mathbb{N}$, the algorithm constructs a W-bounding set.*

Proof. To show property 1, assume for contradiction that $w(y \prec X) > W$ for some y at the end of the algorithm. Consider the maximal subtree of T which contains y and does not contain nodes from X as inner nodes. Let $u \in X \cup \{s_0\}$ be the root of this subtree, and $v \notin X$ be its son in the subtree. At the time where the above test was executed for the edge (u, v) the result was $w(y \prec X' + u) \leq W$. Since $X' + u \subseteq X + s_0$ we have also $w(y \prec X) = w(y \prec X + s_0) \leq w(y \prec X' + u) \leq W$, which contradicts the premise.

Property 2 is immediate from the construction of the test, since it can be observed that, after the test for a node v has been performed, no more nodes from the subtree T_v are later added to X. □

Theorem 3.3 (*Discrete $(1, p)$-Centroid on a Tree*). *A discrete $(1, p)$-centroid on a tree can be found in time $O(n^2 (\log n)^2 \log w(T))$.*

Proof. We perform a binary search to find the smallest weight $W \in [0, w(T)]$ such that there is a W-bounding set X with at most p elements. By Lemmas 3.1 and 3.2, the set found by this approach has follower gain $w_{1,r}^*$ and is therefore a $(1, p)$-centroid.

A straightforward implementation would compute a $(1, X)$-medianoid in the current subtree below each single edge. Using the algorithm from [19], this yields the proposed running time. □

3.2. Absolute $(1, p)$-centroid on a tree

In order to solve the problem in the absolute case, we attempt to discretize the instance, i.e., we show that one can assume that the leader chooses his position always on a finite grid projected onto the edge set. This allows us to reduce the absolute case to the discrete case discussed above.

Theorem 3.4 (*Discretization*). *Let I be an instance of the absolute (r, p)-centroid problem on an arbitrary graph with edge lengths in $\mathbb{N}$. Then there is an (r, p)-centroid X of I such that $d(x, v) \in \frac{1}{2}\mathbb{N}$ for each $x \in X$ and each vertex v.*

Proof. We assume w.l.o.g. that all edges have unit length, which can be achieved by creating zero weighted nodes at an integer grid.

Now let X_p be an (r, p)-centroid. A point z is called (v, X_p)-*isodistant* [17] if there is a node v such that $d(v, z) = d(v, X_p)$. (v, X_p)-isodistant points are of particular importance: they are the only candidates for boundary points of the connected set of all points where the follower claims the node v. Hence the gain of the follower is constant within each interval limited by isodistant points.

We transform X_p into a new set X_p' by moving each point to the nearest node, unless the point is the midpoint of an edge. Notice that each point moves by less than $\frac{1}{2}$ by this transformation. Moreover, all isodistant points also move by less than $\frac{1}{2}$.

We show that $w_r^*(X_p') \leq w_r^*(X_p)$. Assume the contrary. Then there must be an interval between two isodistant points induced by X_p' where the follower gains a set of nodes which was not present in the original instance. This means that

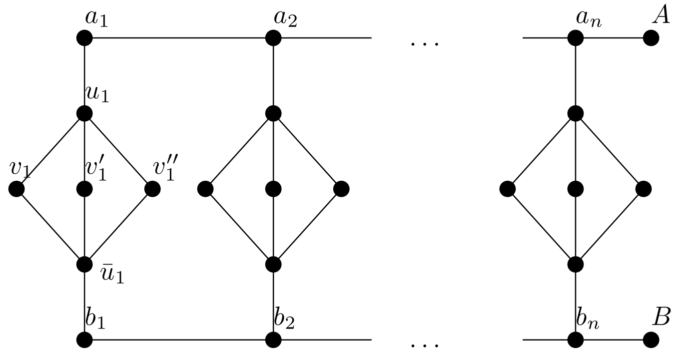

Fig. 5. The discrete $(1, p)$-centroid is NP-hard on a pathwidth bounded graph.

there must be a pair (i_1, i_2) of two isodistant points on an edge which has interchanged its relative position during the transformation. More exactly, let i_1, i_2 be the distances of the points to one fixed endpoint of the edge before the transformation, and i_1', i_2' the positions after the transformation; then we must have $i_1 \geq i_2$ and $i_1' < i_2'$. Obviously i_1', i_2' are either endpoints or midpoints, i.e., $i_1', i_2' \in \{0, \frac{1}{2}, 1\}$ (where for the sake of an easier presentation we identify points with their respective distances).

If one of those points, say i_1', is a midpoint then the point has not moved at all, i.e., $i_1 = i_1'$. This implies that point i_2 has moved by at least $\frac{1}{2}$, which is impossible. On the other hand, if both i_1', i_2' are endpoints, the total sum of the movement is at least 1, which is again a contradiction. This shows the claim. □

We point out that from this result one can only derive that the positions of the leader are discretized to positions in $\frac{1}{2}\mathbb{N}$, while the positions of the follower are still unrestricted.

A direct application of the above result to the algorithm stated in the previous section would yield a new instance where the node number and thus the running time of the algorithm would no longer necessarily be polynomially bounded. Hence we propose a modification of the previous algorithm.

We start the algorithm on the unaltered input tree. Whenever in the original algorithm there is a test on an edge (u, v) to be performed, we now essentially have to determine a point on that edge which is W-bounding. By the above discretization result it turns out that it is sufficient to restrict the tests to (exponentially many) discrete points on that edge. Since all those sub-edges are threaded on the original edge, the interesting point which is W-bounding can be found by a binary search without actually creating all those points as real nodes. This shows the following result:

Corollary 3.5 (*Absolute* $(1, p)$*-Centroid on a Tree*). *An absolute* $(1, p)$*-centroid on a tree can be found in time* $O(n^2 (\log n)^2 \log w(T) \log D)$ *where* $D := \max_e d(e)$.

Proof. The running time follows from similar arguments as above. Notice that the absolute $(1, X)$-medianoid can be computed in $O(n (\log n)^2)$ [12]. □

3.3. Discrete $(1, p)$*-centroid on a pathwidth bounded graph*

In this section we oppose the positive results for the $(1, p)$-centroid on trees with a hardness result for a slightly more complex graph class, namely the class of pathwidth bounded graphs. A *path-decomposition* of a graph (V, E) is a path with node set V' and a mapping $p\colon V \to 2^{V'}$ such that $p(v)$ is a path for all nodes $v \in V$ and $p(v_1) \cap p(v_2) \neq \emptyset$ for all edges $(v_1, v_2) \in E$. The *width* of the decomposition is $\max_{v' \in V'} \left|\{ v \in V \mid p(v) \ni v' \}\right| - 1$. The *pathwidth* of a graph is the minimum width of a path decomposition.

Theorem 3.6 (*Hardness on Pathwidth Bounded Graphs*). *Determining a discrete or an absolute* $(1, p)$*-centroid on a pathwidth bounded graph is NP-hard.*

Proof. Let an instance of problem Partition be given as in Theorem 1.1. Construct a graph as follows (see Fig. 5): Start with two paths a_1–a_2–…–a_n–A and b_1–b_2–…–b_n–B. For each $i = 1, \ldots, n$, add a connecting path a_i–u_i–v_i–$\bar{u}_i$–b_i and complement it by u_i–v_i'–$\bar{u}_i$ and u_i–v_i''–$\bar{u}_i$ to form a diamond. All edges have unit length except for the edges on the initial a-path and b-path which have length $< \frac{1}{n}$. The node weights are set to $w(u_i) := w(\bar{u}_i) := s_i$ and $w(v_i) := w(v_i') := w(v_i'') = \Omega$ for an $\Omega > S^*$. The weights of the a_i, b_i nodes are set to 1, and finally $w(A) := w(B) := \Omega + 1$.

We claim: For $p := n$ there is a discrete $(1, p)$-centroid of weight $W := \frac{1}{2}S^* + n + \Omega + 1$ if and only if the Partition instance is solvable. (The proof for the absolute case is identical.)

"If": Let $S' \subset S$ be a subset with $\sum S' = \frac{1}{2}S^*$. For each $i = 1, \ldots, n$ place the leader at u_i if $s_i \in S'$ and at $\bar{u}_i$ otherwise. The follower places a server at B and claims all b-nodes, plus those nodes $\bar{u}_i$ where $s_i \in S'$ which results in a total gain of W.

	(r, p)-centroid *(the leader problem)*		(r, X_p)-medianoid
	absolute	discrete	*(the follower problem)*
arb. r arb. p	NP-hard on path *[Theorem 2.1]*	$O(pn^4)$ on path *[Theorem 2.4]* NP-hard on spider *[Theorem 2.5]* Σ_2^p-complete on graph *[15]*	$O(n)$ on path *[14]* $O(rn^2)$ on tree *[14]* NP-hard on graph *[14]*
$r = 1$ arb. p	$O(n^2(\log n)^2 \log W \log D)$ on tree [Corollary 3.5]	$O(n^2 (\log n)^2 \log W)$ on tree *[Theorem 3.3]* NP-hard on pathwidth bounded graph *[Theorem 3.6]*	$O(n (\log n)^2 / \log \log n)$ on tree *[19]* $O(n^2 \log n + nm)$ on graph *[by enumeration]*
$r = 1$ $p = 1$	$O(n^4 m^2 \log mn \log W)$ on graph *[11]*	$O(n^3)$ on graph *[4]*	

Fig. 6. Complexity of the (r, p)-centroid problem. $W := \sum w(v)$ and $D := \max d(e)$. The hardness results from the discrete case also apply to the absolute case.

"Only if": Consider diamond i. If the leader places no server, the follower could claim more than 3Ω. Hence there must be one server per diamond. If the leader places a server at a v-node, the follower could still claim more than 2Ω. As a consequence, the leader places either at u_i or at $\bar{u}_i$. Let $S' := \{ s_i \mid \text{the leader places server at } u_i \}$.

The follower can not claim two or more v-nodes with a single server. Hence it is optimal to place on A or B which claims a fixed weight of $\Omega + 1 + n$, plus the weight $\sum S'$ (if the follower places a server at B) or $S^* - \sum S'$ (if the follower places a server at A). If $\sum S' \neq \frac{1}{2} S^*$ this is larger than W.

The proof is completed by the observation that the constructed graph has pathwidth 7. □

4. Conclusions

Fig. 6 provides an overview on the complexity status of the (r, p)-centroid problem, i.e., the problem of optimally placing the leader. For completeness we have added the known results for the corresponding follower problem variants. Notice that in the follower problem we do not distinguish between the absolute model and the discrete model since the complexity is the same in both cases.

In [17], the authors approach the absolute (r, X_p)-medianoid problems by *polynomial discretization*, i.e., in the infinite set of points they identify polynomially many points and this way reduce the absolute problem to a finite discrete problem. Since we have shown that on a path the absolute (r, p)-centroid is NP-hard while the discrete is not, we conjecture that such a polynomial discretization is unlikely to work for the absolute (r, p)-centroid problem in general. (Notice that the discretization employed in Section 3.2 is not polynomial.)

There are a few further problems left open at this point. First, the purpose of the current paper is to distinguish NP-hard from polynomial time solvable problem instances, and it would be not surprising if the algorithms we propose here can be improved in running time. Second, in [15] it has been shown that the (r, p)-centroid can not be approximated within a factor of $n^{1-\varepsilon}$ on general graphs; in [18] there has been provided an FPTAS for the absolute (r, p)-centroid on paths. In connection with the hardness results in this paper, approximability on trees and other graph classes is worth investigating.

Acknowledgements

We thank the anonymous referee for detailed and helpful comments, and Arie Tamir for pointing us to [12] which implies a faster running time in Corollary 3.5.

References

[1] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, M. Protasi, Complexity and Approximation, Springer, Berlin, Heidelberg, New York, 1999.
[2] S. Benati, NP-hardness of some competitive location models with probabilistic choice rules, Studies in Locational Analysis 14 (2000) 211–231.
[3] T.H. Cormen, C.E. Leiserson, R.L. Rivest, Introduction to Algorithms, MIT Press, 1990.
[4] C.M. Campos Rodríguez, J.A. Moreno Pérez, Relaxation of the Condorcet and Simpson conditions in voting location, European Journal of Operational Research 145 (2003) 673–683.
[5] C.M. Campos Rodríguez, J.A. Moreno Pérez, Multiple voting location problems, European Journal of Operational Research 191 (2) (2008) 437–453.
[6] H.A. Eiselt, G. Laporte, Sequential location problems, European Journal of Operational Research 96 (1996) 217–231.
[7] M.R. Garey, D.S. Johnson, Computers and Intractability (A Guide to the Theory of NP-Completeness), W.H. Freeman and Company, New York, 1979.
[8] A.J. Goldman, Optimal center location in simple networks, Transportation Science 5 (1971) 212–221.

[9] S.L. Hakimi, On locating new facilities in a competitive environment, European Journal of Operational Research 12 (1983) 29–35.
[10] S.L. Hakimi, Locations with spatial interactions: Competitive locations and games, in [13], 1990, pp. 439–478.
[11] P. Hansen, M. Labbé, Algorithms for voting and competitive location on a network, Transportation Science 22 (4) (1988) 278–288.
[12] Tae Ung Kim, Timothy J. Lowe, Arie Tamir, James E. Ward, On the location of a tree-shaped facility, Networks 28 (3) (1996) 167–175.
[13] P.B. Mirchandani, R.L. Francis, Discrete Location Theory, in: Series in Discrete Mathematics and Optimization, Wiley-Interscience, 1990.
[14] N. Megiddo, E. Zemel, S. Hakimi, The maximum coverage location problem, SIAM Journal on Algebraic and Discrete Methods 4 (2) (1983) 253–261.
[15] H. Noltemeier, J. Spoerhase, H.-C. Wirth, Multiple voting location and single voting location on trees, European Journal of Operational Research 181 (2007) 654–667.
[16] A.P. Punnen, Y.P. Aneja, On k-sum optimization, Operations Research Letters 18 (1996) 233–236.
[17] D.R. Santos-Peñate, R. Suárez-Vega, P. Dorta-González, The leader-follower location model, Networks and Spatial Economics 7 (2007) 45–61.
[18] J. Spoerhase, H.-C. Wirth, Approximating (r, p)-centroid on a path, in: Proceedings of the 7th Cologne-Twente Workshop on Graphs and Combinatorial Optimization, CTW'08, 2008. Available at: http://ctw08.dti.unimi.it/CTW08-Proceedings.pdf.
[19] J. Spoerhase, H.-C. Wirth, An $O(n(\log n)^2/\log\log n)$ algorithm for the single maximum coverage location or the $(1, X_p)$-medianoid problem on trees, Information Processing Letters 109 (8) (2009) 391–394.