Available online at www.sciencedirect.com
SCIENCE DIRECT®

Computational Geometry 27 (2004) 193–210

Computational Geometry
Theory and Applications

www.elsevier.com/locate/comgeo

Median problem in some plane triangulations and quadrangulations

Victor Chepoi *, Clémentine Fanciullini, Yann Vaxès

Laboratoire d'Informatique Fondamentale, Université de la Méditerranée, Faculté des Sciences de Luminy, F-13288 Marseille Cedex 9, France

Received 3 June 2002; received in revised form 15 July 2003; accepted 13 November 2003

Communicated by P. Widmayer

Abstract

In this note, we present linear-time algorithms for computing the median set of plane triangulations with inner vertices of degree $\geqslant 6$ and median vertices of plane quadrangulations with inner vertices of degree $\geqslant 4$.

Keywords: Plane triangulation; Median problem; Distance

1. Introduction

Given a finite, connected graph $G = (V, E)$ endowed with a non-negative weight function $\pi(v)$ $(v \in V)$ the *median set* $\mathrm{Med}(\pi)$ consists of all vertices x minimizing the total weighted distance

$$F_\pi(x) = \sum_{v \in V} \pi(v) d(v, x).$$

Finding the median set of a graph, or, more generally, of a network or a finite metric space is a classical optimization and algorithmic problem with many practical applications. The weighted version of the median problem is one of the basic models in facility location (where it is sometimes called the Fermat–Weber problem); see, for example, [25]. It arises with majority consensus in classification and data analysis [4,8,23], where the median points are usually called Kemeny medians. Algorithms for locating medians in graphs are especially useful in the areas of transportation and communication in distributed networks: placing a common resource at a median minimizes the cost of sharing the resource

* Corresponding author.
E-mail address: chepoi@lidil.univ-mrs.fr (V. Chepoi).

0925-7721/$ – see front matter
doi:10.1016/j.comgeo.2003.11.002

with other locations or the total time of broadcasting messages. Recently, motivated by a heuristic for reconstructing discrete sets from projections presented in [7], the median sets of polyominoes and some other special subsets of the square grid have been investigated in [19,22] (in [17], similar questions have been considered in the more general setting of linear metrics). Finally, [11] provides efficient algorithms for the approximate computation of the values of the function $F_\pi(x)$ for points in $\mathbb{R}^n$ endowed with the Euclidean distance.

Quite a few algorithms are known for finding medians of graphs (see [25] for an overview), but only for trees the classical *majority rule* yields a linear-time algorithm [20,21,28]: given a tree T and an edge $e = xy$, the median set is contained in the heaviest of two subtrees T_x, T_y defined by this edge (if the subtrees have the same weight, then both x and y are medians). This is so because $F_\pi(x) - F_\pi(y)$ equals the difference between the weights of T_y and T_x and every local minimum of the function F_π is a global minimum; for more details, see [25]. Later, using techniques from computational geometry this approach has been extended in [15] and an efficient algorithm for the median problem on simple rectilinear polygons with an intrinsic l_1-metric has been designed. More recently, similar ideas were used in [2,13] to develop simple (but nice) self-stabilizing algorithms for finding medians of trees; see also [1,3] for efficient algorithms for maintaining medians in dynamic trees. Last but not least, [6] characterizes the graphs in which all local medians are (global) medians for each weight function π (by a *local median* one means a vertex x such that $F_\pi(x)$ does not exceed $F_\pi(y)$ for any neighbor y of x). In particular, it is shown in [6] that these graphs can be recognized in polynomial time and that they are exactly the graphs in which all median sets induce connected or isometric subgraphs.

In this note, we describe linear-time algorithms for computing the median sets in two classes of face regular plane graphs. Namely, we consider plane triangulations with inner vertices of degree at least six (called *trigraphs*) and plane quadrangulations with inner vertices of degree at least four (called *squaregraphs*); see Fig. 1 for examples. Particular cases of these graphs are the subgraphs of the regular triangular and square grids which are induced by the vertices lying on a simple circuit and inside the region bounded by this circuit (the latter comprises the graphs from [19,22]). Notice that these classes of plane graphs are particular instances of bridged and median graphs, two classes of graphs playing an important role in metric graph theory. The trigraphs have been introduced and investigated in [5] where they are

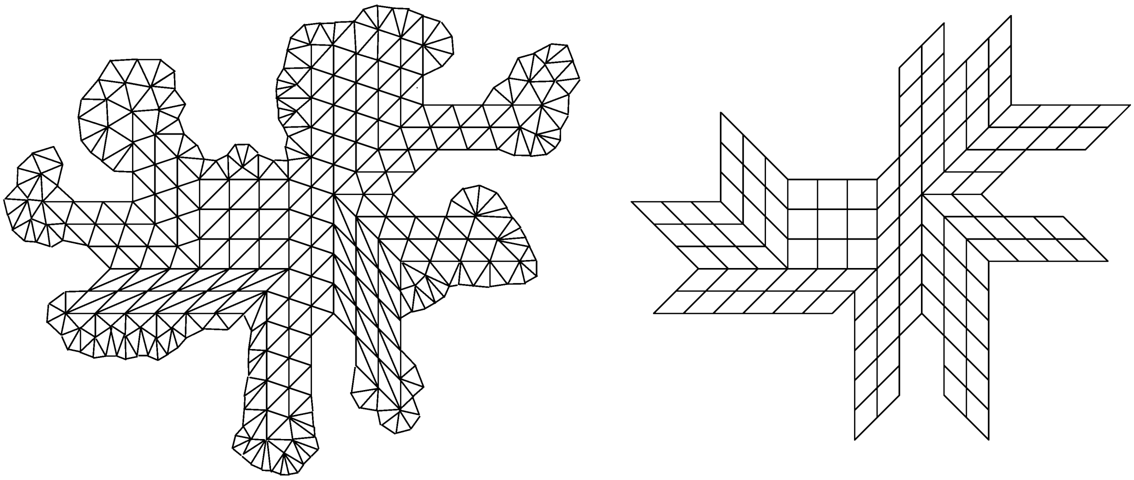


Fig. 1. Examples of trigraphs and squaregraphs.

the basic building stones in the construction of weakly median graphs. The present paper continues the line of research of [16] where linear-time algorithms for computing the diameter and the center of these graphs have been proposed (the terms "trigraph" and "squaregraph" are from that paper). It should be noted that, due to the different nature of the objective functions (minsum and minmax), the method used here is completely different from that of [16]. Nevertheless, both papers have the same flavor of developing a kind of computational geometry in plane graphs based on natural convexity and metric properties of graphs in question.

By replacing every inner face of a plane triangulation by an equilateral triangle of side 1, one obtains a two-dimensional (pseudo-)manifold which can be embedded in some high dimensional space. Analogously, one can define such a manifold if one replaces every inner face of a plane quadrangulation by a unit square. (For squaregraphs and trigraphs such manifolds can be effectively constructed via an isometric embedding of these graphs into hypercubes and half-cubes as is done in [5].) The resulting manifolds can be endowed in a natural way with an intrinsic Euclidean metric as is explained in [12]. Now, the trigraphs and the squaregraphs are precisely the plane triangulations and quadrangulations for which these surfaces have intrinsic metric of non-positive curvature [9,12], therefore they may arise, among others, in the following type of applications. Recently, [10,26] proposed a new technique (called Isomap) of data analysis (as an alternative to principal component analysis or multidimensional scaling) which, on the base of easily measured local metric information, aims to build the underlying global geometry of data sets in the form of a low-dimensional structure in a high-dimensional data space. Isomap deals with data sets of $\mathbb{R}^n$ which are assumed to lie on a smooth manifold M of low dimension. The crucial stage of the method consists in approximating the unknown geodesic distance in M between data points in terms of the graph distance with respect to some graph G constructed on the data points. Hence, when M is 2-dimensional and has non-positive curvature it is likely that the resulting graph G will be a trigraph or a squaregraph. On the other hand, the medians computed with respect to the distance function of G can be viewed as a natural extension of the usual notion of median used in data analysis and statistics. Finally, notice that terrains can be viewed as particular instances of such pseudo-manifolds.

Our method of computing $\mathrm{Med}(\pi)$ for trigraphs is based on the following. From the results of [6] it follows that in trigraphs the function F_π is unimodal for all the choices of weights. Unlike for trees, this fact alone does not yield a linear-time algorithm because computing $F_\pi(x)$ for a single vertex x already needs linear time. Instead, for trigraphs we show how to compute in total linear time the differences $\Delta(x,y) := F_\pi(x) - F_\pi(y)$ for all edges xy of G. Using this information, we define the directed graph $\vec{G}_\pi$ in which the edge xy of G is replaced by the arc $\overrightarrow{yx}$ if $\Delta(x,y) < 0$ and by the arc $\overrightarrow{xy}$ if $\Delta(x,y) > 0$; no arc between x and y is defined if $\Delta(x,y) = 0$. Due to the unimodality of the function F_π, the median set $\mathrm{Med}(\pi)$ consists of all sinks (vertices having no outgoing arcs) of the resulting acyclic graph $\vec{G}_\pi$. Clearly, these vertices can be found in linear time by traversing $\vec{G}_\pi$. Notice also that with these differences in hand we can easily compute in total linear time all values of the function F_π in the following way: compute $F_\pi(c)$ for some vertex c and construct a tree rooted at c using any graph traversal. For each vertex v let v' be its father in this tree. Now, if $F_\pi(v')$ has been already computed, then set $F_\pi(v) := F_\pi(v') + \Delta(v,v')$, and continue the traversal of the tree. For squaregraphs, the majority rule together with a simple trick yield a divide-and-conquer linear-time algorithm for computing a part of the median set $\mathrm{Med}(\pi)$.

The paper is organized as follows. In the next section, we recall some necessary notions and formulate some auxiliary results. In Section 3 we present the algorithm for computing medians of squaregraphs. In Section 4 we describe the main contribution of this note–a linear-time algorithm for the median problem in trigraphs.

2. Preliminaries

All graphs $G = (V, E)$ occurring in this note are connected, finite and undirected. Since computing the median set of a graph can be reduced in linear time to computing the median sets inside its 2-connected components [21], we may assume without loss of generality that G itself is 2-connected. In a graph G, the *length* of a path from a vertex v to a vertex u is the number of edges in the path. The *distance* $d(u, v)$ from u to v is the length of a minimum length (u, v)-path and the *interval* $I(u, v)$ between these vertices is the set $I(u, v) = \{w \in V\colon\ d(u, v) = d(u, w) + d(w, v)\}$. A subset (or the subgraph induced by this subset) $S \subseteq V$ is called *convex* if $I(u, v) \subseteq S$ whenever $u, v \in S$, and *gated* [21] if for each $v \notin S$ there exists a (necessarily unique) vertex $v' \in S$ (the *gate* of v in S) such that $v' \in I(v, u)$ for every $u \in S$ (notice that gated sets are convex). By a *half-plane* of G we will mean a convex set H with a convex complement $V - H$. For a weight function π and a subset S of vertices, let $\pi(S) = \sum_{s\in S} \pi(s)$ denote the weight of S. In particular, $\pi(V)$ denotes the total weight of vertices of G. Obviously, we can suppose that $\pi(V)$ is known in advance (otherwise, it can be easily computed in linear time).

For an edge uv of a graph G, let

$$W(u, v) = \bigl\{x \in V\colon\ d(u, x) < d(v, x)\bigr\}.$$

The following well-known lemma is trivial but crucial.

Lemma 1. *For every weight function π and every edge uv of G we have*

$$F_\pi(u) - F_\pi(v) = \pi\bigl(W(v, u)\bigr) - \pi\bigl(W(u, v)\bigr).$$

Indeed, a vertex x of $W(v, u)$ contributes with $+\pi(x)$ to $F_\pi(u) - F_\pi(v)$, a vertex x of $W(u, v)$ contributes with $-\pi(x)$ to this difference, while every vertex equidistant to u and v does not contribute at all. Summing over all vertices of G, we obtain the right-hand side.

In view of Lemma 1, in order to construct the oriented graph $\vec{G}_\pi$ efficiently, we must be able to compute $\pi(W(u, v))$ and $\pi(W(v, u))$ for all edges uv of G. If G is a bipartite graph, then $W(u, v) \cup W(v, u) = V$, therefore it is enough to find the weight of only one of these complementary sets. Moreover, if G is a squaregraph, then $W(u, v)$ and $W(v, u)$ are gated sets, because the squaregraphs are median graphs; cf. [27]. From the results of [21] follows that in this case $\mathrm{Med}(\pi) \subseteq W(u, v)$ if and only if $\pi(W(u, v)) > \pi(W(v, u))$. In case of trigraphs, the sets $W(u, v)$ and $W(v, u)$ are convex, but they no longer cover the whole vertex-set of G. Nevertheless, these sets extend to two pairs of complementary half-planes. In Section 4 we will show that the half-planes of trigraphs have a geometric nature which allows to process them efficiently.

To conclude this section, notice that in subsequent algorithms every trigraph or squaregraph G is represented by a *doubly-connected edge list*; for precise definition and details see [18]. We recall here only a few things about this data structure. Since every edge of G bounds two faces, it is convenient to view the different sides of an edge as two distinct *half-edges*. The two half-edges $\overrightarrow{xy}$ and $\overrightarrow{yx}$ we get for an edge xy are called *twins* (so that *twin*$(\overrightarrow{xy}) = \overrightarrow{yx}$ and *twin*$(\overrightarrow{yx}) = \overrightarrow{xy}$). The half-edges bounding the outer face ∂G are oriented so that ∂G is traversed in clockwise order. On the other hand, the half-edges of every inner face are oriented so that the face is traversed in counterclockwise order. The half-edge record of a half-edge $\vec{e}$ stores a pointer to its origin, a pointer to its twin, a pointer to the incident face, and two pointers *next*$(\vec{e})$ and *prev*$(\vec{e})$ to the next and the previous edges on the boundary of incident face.

3. Computing median sets in squaregraphs

From the results of [14] follows that every squaregraph G is a median graph, i.e., for any three vertices x, y and z of G there exists a unique vertex which is simultaneously on shortest (x, y)-, (y, z)- and (x, z)-paths. Next we specify some known properties of median graphs and their median sets to the case of squaregraphs; cf. [4,27] and the references therein. In subsequent results, G is a squaregraph and uv is an edge of G.

(S1) $W(u, v)$ and $W(v, u)$ are gated and constitute a pair of complementary half-planes.

Let P be the subgraph of G induced by all vertices of $W(u, v)$ having a neighbor in the set $W(v, u)$ (analogously one can define the subgraph $Q \subseteq W(v, u)$). For every vertex $x \in P$, let F_x consists of all vertices of $W(v, u)$ whose gate in $W(u, v)$ is the vertex x and call this set the *fiber* of x. Analogously define the fiber F_y of every vertex $y \in Q$.

(S2) P and Q are gated paths of G. Additionally, all fibers F_x, $x \in P \cup Q$, are gated.

Notice that there is a natural isomorphism between the paths P and Q. For all edges $u'v'$ with $u' \in P$ and $v' \in Q$ one has $W(u', v') = W(u, v)$ and $W(v', u') = W(v, u)$. The subgraph induced by $P \cup Q$ is a strip consisting of one or several inner faces of G. This strip and the paths P and Q can be easily constructed in $O(|P| + |Q|)$ time starting from an edge uv on the outer face of G and the unique inner face containing this edge (using a similar procedure as in the case of strips in trigraphs).

We continue with some properties of median sets in squaregraphs. From (S1), (S2), and the general result of [21], we deduce the following *majority rule* for squaregraphs [4,24]:

(S3) The median set $\mathrm{Med}(\pi)$ is gated. Moreover, $\mathrm{Med}(\pi)$ is contained in $W(u, v)$ if $\pi(W(u, v)) > \frac{1}{2}\pi(V)$ and $\mathrm{Med}(\pi)$ is contained in $W(v, u)$ if $\pi(W(v, u)) > \frac{1}{2}(\pi(V))$. Finally, if $\pi(W(u, v)) = \pi(W(v, u))$, then $\mathrm{Med}(\pi)$ intersects both gated paths P and Q.

Therefore we can continue the search in the subgraph induced by $W(u, v)$ in the first case, in the subgraph induced by $W(v, u)$ in the second case, and in the strip $P \cup Q$ in the third case. The respective subgraph G' is endowed with a new weight function π' defined in the following way. In the first case, define π' on $W(u, v)$ by setting $\pi'(x) := \pi(x)$ for every $x \in W(u, v) - P$ and $\pi'(x) := \pi(F_x) + \pi(x)$ for every $x \in P$. Analogously, in the second case define π' on $W(v, u)$ by setting $\pi'(y) := \pi(y)$ for every $y \in W(v, u) - Q$ and $\pi'(y) := \pi(F_y) + \pi(y)$ for every $y \in Q$. Finally, in the third case define π' on $P \cup Q$ by setting $\pi'(x) := \pi(F_y)$ and $\pi'(y) = \pi(F_x)$, where $x \in P$ and $y \in Q$ are adjacent to each other. Then one can see that $\mathrm{Med}(\pi') = \mathrm{Med}(\pi)$ in first and second cases and that $\mathrm{Med}(\pi') \subseteq \mathrm{Med}(\pi)$ in third case. In the latter case $\mathrm{Med}(\pi')$ can be easily computed applying the majority rule to the resulting strip.

In order to implement this algorithm in linear time, at each step we have to decide in which case of (S3) we are by traversing a part of the current graph G proportional in size to the half-plane which will be removed from further consideration. For example, if $\mathrm{Med}(\pi)$ is contained in $W(u, v)$, then we have to decide this in time $O(|W(v, u)|)$. This is possible using the following procedure: perform simultaneously the Breadth-First-Search on the sets $W(u, v)$ and $W(v, u)$ starting from the paths P and Q, respectively, and stop when one of the sets will be completely traversed. (This can be easily done by alternatively searching each of the half-planes according to BFS.) During these BFS traversals, we add the weight of the current vertex v to the weight of the half-plane and the fiber F_x containing it. For this notice that v will be in the same half-plane and the same fiber as its father in the respective BFS tree. Suppose without loss of generality that the search of $W(v,u)$ was completed first. If $\pi(W(v,u)) < \frac{1}{2}\pi(V)$, then $\mathrm{Med}(\pi)$ is contained in $W(u,v)$ and we spent $O(|W(v,u)|)$ time to decide this and to construct the weight function π'. Since finding the median set in $W(u,v)$ will take $O(|W(u,v)|)$ time, we conclude that the overall time is $O(|V|)$. On the other hand, if $\pi(W(v,u)) \geqslant \frac{1}{2}\pi(V)$, then $\mathrm{Med}(\pi)$ is contained in $W(v,u)$. In this case, we continue the traversal of $W(u,v)$ in order to compute the weights of all fibers of this set. Since $|W(u,v)| \geqslant |W(v,u)|$, we spent $O(|W(u,v)|)$ time to conclude that the search of median vertices should be continued in $W(v,u)$ or in $P \cup Q$. All this shows that employing this simple approach we can find at least one part of $\mathrm{Med}(\pi)$ in linear time. If the weights of all vertices are positive, then $\mathrm{Med}(\pi)$ is either a vertex, an edge or a square [4,24], therefore our algorithm will return the whole median set. For arbitrary non-negative weight functions, to compute the whole median set either we have to expand the computed part in a careful way by taking into account that $\mathrm{Med}(\pi)$ is an interval [4] or to adopt an approach similar to that for trigraphs presented in the next section.

4. Computing median sets in trigraphs

Throughout this section, $G=(V,E)$ is a trigraph stored in the form of a doubly-connected edge list whose outer face is traversed clockwise. Notice that the outer face of each other type of regions occurring below (half-planes, sectors, cones and strips) is also traversed clockwise. The *ball* $B_r(c)$ of radius r and center c consists of all vertices at distance at most r from c. The *neighborhood* $N(S)$ of a set S consists of S and all vertices of $V - S$ having a neighbor in S. We recall some properties of trigraphs established in [5].

(T1) The balls and the neighborhoods of convex sets of trigraphs are convex.

From this property one can easily conclude that trigraphs do not contain induced 4- and 5-cycles.

(T2) Trigraphs do not have induced subgraphs isomorphic to the 4-clique K_4 and the graph $K_{1,1,3}$ consisting of three triangles having an edge in common.

From the definition and these properties immediately follows that the subgraphs induced by the convex sets of a trigraph are also trigraphs.

(T3) If two adjacent vertices x, y of a trigraph are equidistant from a vertex v, then there exists a common neighbor of x and y one step closer to v.

As established in [5], trigraphs contain a rich amount of *half-planes* (convex sets with convex complements):

(T4) For any two adjacent vertices u and v of a trigraph G there exist exactly two distinct pairs of complementary half-planes H'_u, H'_v and H''_u, H''_v separating u and v, i.e., such that $u \in H'_u \cap H''_u$ and $v \in H'_v \cap H''_v$. These half-planes satisfy the equalities $W(u,v) = H'_u \cap H''_u$ and $W(v,u) = H'_v \cap H''_v$.

4.1. Half-planes, sectors, cones, zips

Let $(H_1, H'_1), \ldots, (H_m, H'_m)$ be the pairs of complementary half-planes of G. Denote by P_i and P'_i the subgraphs induced by the vertices of H_i and H'_i which have neighbors in the complementary half-plane (i.e., in H'_i and H_i, respectively).

Lemma 2. *Each of the subgraphs P_i and P'_i is a convex path of G having both end-vertices on the outer face ∂G.*

Proof. P_i is the intersection of two convex sets H_i and $N(H'_i)$, therefore it is convex (analogously one deduces that P'_i is convex). Since G is K_4-free and P'_i is convex, every vertex of P_i has one or two adjacent neighbors in P'_i.

We assert that two adjacent vertices x, y of P_i have a common neighbor in P'_i. Pick $x', y' \in P'_i$ such that x' is adjacent to x and y' is adjacent to y, and suppose that $x' \neq y'$. Since H'_i is convex, $d(x', y') \leqslant 2$. If x' and y' are adjacent, then we obtain a 4-cycle (x, y, y', x') which cannot be induced. Therefore one of the vertices x', y' is a common neighbor of x and y. On the other hand, if $d(x', y') = 2$, then pick a common neighbor z' of x' and y'. It necessarily belongs to $I(x', y') \subseteq P'_i$. The 5-cycle (x, y, y', z', x') cannot be induced, whence z' is adjacent to x and y, thus establishing our assertion.

P_i is a convex subgraph of G, thus it is also a trigraph. Therefore to show that P_i is a path it suffices to prove that it does not contain 3-cycles and vertices of degree 3. Suppose by way of contradiction, that P_i contains three pairwise adjacent vertices x, y, z. If they have a common neighbor in P'_i we will get a K_4, a contradiction with (T2). So let $z' \in P'_i$ be the common neighbor of x and y, $y' \in P'_i$ be the common neighbor of x and z, and $x' \in P'_i$ be the common neighbor of y and z. The vertices x', y' and z' are pairwise adjacent because P'_i is convex. Now, in order to avoid an induced 4-cycle generated by x, y, x', y', either x and x' are adjacent or y and y' are adjacent. In both cases, we obtain a 4-clique, which is impossible. Thus P_i and P'_i induce acyclic subgraphs. Finally, assume by way of contradiction that P_i contains a $K_{1,3}$, i.e., a vertex x adjacent to three other vertices y, z, v. Now, if we consider the common neighbors y', z', v' in P'_i of y and x, z and x, and v and x, respectively, the convexity of P_i and P'_i implies that these vertices must be distinct and pairwise adjacent. This contradicts the fact that P'_i does not contain 3-cycles, therefore indeed P_i and P'_i are convex paths.

Finally, pick a vertex u of P_i which is an inner vertex of G. Since H'_i is convex, u has only two (adjacent) neighbors v', v'' in H'_i (and P'_i). The neighbors of v' and v'' from $N(u)$ belong to P_i, hence u is also an inner vertex of P_i. Therefore the end-vertices of P_i belong to ∂G. □

We call the convex paths P_i and P'_i the (*border*) *lines* of the half-planes H_i and H'_i. Denote by Z_i the partial subgraph of G comprising all edges with one end in P_i and another one in P'_i and, due to its form, call Z_i a *zip*; see Fig. 2 for an illustration. A *strip* S_i is the union of all inner faces of G sharing two edges with the zip Z_i. Notice that every zip Z_i shares two edges $e_i = uv$ and $e'_i = u'v'$ with ∂G, so that S_i lies to the left of the half-edge of e_i which bounds ∂G. Below we will show that the zip Z_i can be reconstructed in a canonical way starting from the half-edge $\vec{e_i}$.

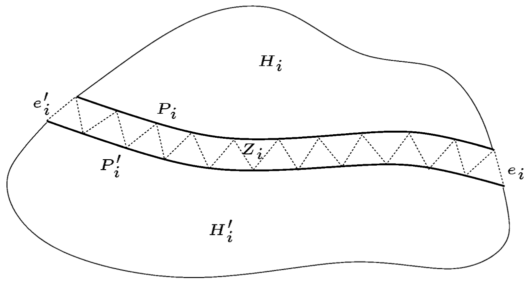


Fig. 2. Half-planes and their border lines.

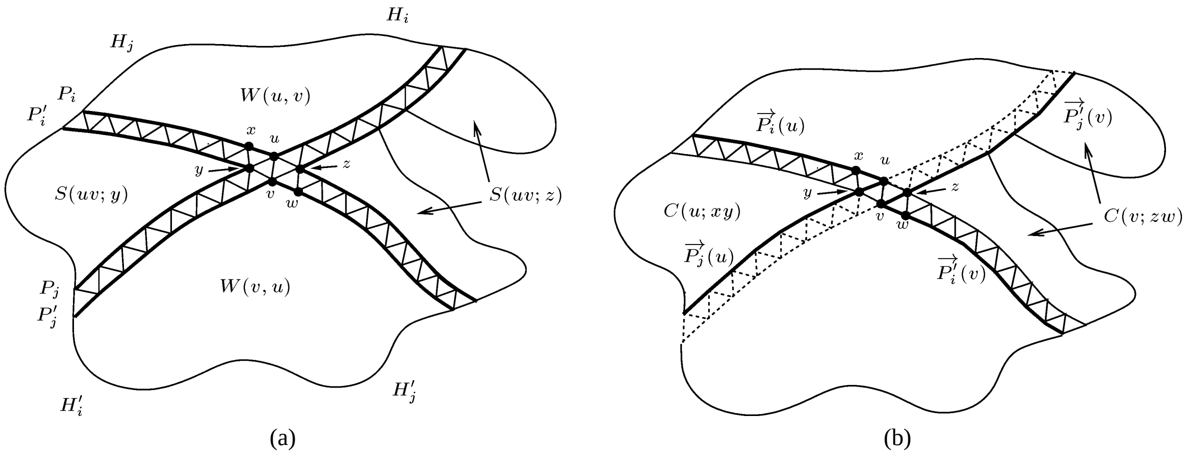


Fig. 3. Sectors and cones.

Let R_i be the region of the plane bounded by the path P_i and the subpath of ∂G comprised between v and u'. Analogously, define the region R'_i of the plane bounded by the path P_i and the subpath of ∂G between the vertices v' and u. Then H_i (respectively H'_i) consists of those vertices of G which are located in R_i (respectively R'_i). Indeed, if a vertex x of H_i belongs to R'_i, then every shortest path between x and a vertex of $P_i \subseteq H_i$ will intersect P'_i, and we get a contradiction with the convexity of H_i.

According to (T4) every two adjacent vertices u and v of G are separated by exactly two distinct pairs of complementary half-planes H_i, H'_i and H_j, H'_j, where $u \in W(u,v) = H_i \cap H_j$ and $v \in W(v,u) = H'_i \cap H'_j$. The two other intersections $H_i \cap H'_j$ and $H'_i \cap H_j$ are called *sectors* and are denoted by $S(uv;y)$ and $S(uv;z)$, respectively, where y and z are the common neighbors of u and v (if the edge uv belongs to the outer face ∂G, then only one of two sectors is defined). From the equalities $H_i = V \cap R_i$ and $H_j = V \cap R_j$ we conclude that $W(u,v)$ consists of all vertices of G located in the region $R_i \cap R_j$. Analogously, $W(v,u) = V \cap R'_i \cap R'_j$, $S(uv;y) = V \cap R_i \cap R'_j$, and $S(uv;z) = V \cap R'_i \cap R_j$. Since $W(u,v) = H_i - S(uv;z)$ and $W(v,u) = H'_j - S(uv;z)$, in order to find the weight of the sets $W(u,v), W(v,u) (uv \in E)$ it suffices to compute the weights of all half-planes and sectors; see Fig. 3(a). To do this, we find more appropriate to perform all computations with objects slightly different from sectors, which we call cones and define below.

Denote by $\overrightarrow{P_i}(u)$ and $\overleftarrow{P_i}(u)$ the sub-paths of the path P_i (with respect to the clockwise traversal of ∂H_i) such that $\overrightarrow{P_i}(u) \cap \overleftarrow{P_i}(u) = \{u\}$ and $\overrightarrow{P_i}(u) \cup \overleftarrow{P_i}(u) = P_i$. Call the oriented paths $\overrightarrow{P_i}(u)$ and $\overleftarrow{P_i}(u)$

u-rays. (Analogously one can define the u-rays $\overrightarrow{P_j}(u)$, $\overleftarrow{P_j}(u)$ and the v-rays $\overrightarrow{P'_i}(v)$, $\overleftarrow{P'_i}(v)$, $\overrightarrow{P'_j}(v)$ and $\overleftarrow{P'_j}(v)$.) Notice that every inner half-edge $\overrightarrow{uv}$ extends to a unique u-ray which we will denote by $\overrightarrow{P}(u,v)$. Let x be the neighbor of u in the ray $\overrightarrow{P_i}(u)$. Then $\overrightarrow{P_i}(u) = \overrightarrow{P}(u,x)$. Set $C(u;xy) := S(uv;y) \cup \overrightarrow{P_i}(u)$ and call the set $C(u;xy)$ a *cone* with *apex* u and *generator* xy, see Fig. 3(b) for an illustration. The u-rays $\overrightarrow{P_i}(u) = \overrightarrow{P}(u,x)$ and $\overrightarrow{P_j}(u) = \overrightarrow{P}(u,y)$ are called the *bounding rays* of $C(u;xy)$. (From what has been established for sectors, $C(u;xy)$ consists of all vertices of the graph G located in the region of the plane bounded by the rays $\overrightarrow{P}(u,x)$ and $\overrightarrow{P}(u,y)$ and a subpath of ∂G comprised between the end-vertices of these rays.) Analogously, if w is the neighbor of v in the ray $\overrightarrow{P'_i}(v)$, we define the cone $C(v;zw) := S(uv;z) \cup \overrightarrow{P'_i}(v)$ with apex v, generator zw and bounding rays $\overrightarrow{P'_i}(v)$ and $\overrightarrow{P'_j}(v)$. In order to deal with degenerated cases, it will be convenient to extend the notion of a cone to the case when $u \in \partial G$ and $x = y \in \partial G$; we denote such a cone by $C(u;xx)$ or $C(u;yy)$ and call it *degenerated*. Notice that a degenerated cone $C(u;xx)$ may be viewed as a usual cone $C(u;xy)$ in the trigraph obtained from G by adding a new vertex y and making it adjacent to two consecutive vertices u, x of ∂G (analogously, $C(u;yy)$ may be viewed as the cone $C(u;xy)$ in the trigraph obtained from G by adding a new vertex x adjacent to u and y). In a similar way one can define the cone $C(u;xy)$ in the case when at least one of the edges ux or uy belong to ∂G: if, say, $uy \in \partial G$, then add a new vertex v adjacent to u and y, and define the sector $S(uv;y)$ and the cone $C(u;xy)$ in the resulting graph. Hence a cone $C(u;xy)$ is defined for every triplet u, x, y of vertices of G, such that u is adjacent to both x, y, and the vertices x, y are adjacent or coincide. In the sequel it suffices to show how to deal with non-degenerated cones only (the degenerated cones do not come from the sectors of the initial graph G, nevertheless they are used in the recursive computation of weights of other cones).

We will establish in Section 4.4 that every half-plane H_i can be represented as a union of cones having their apices at the origin of $\vec{e_i}$, therefore $\pi(H_i)$ (and therefore $\pi(H'_i)$) can be computed provided we know the weights of the cones of G.

4.2. Computing zips, strips, lines, and weights of rays

To perform this computation, we traverse the half-edges of ∂G in clockwise order. Let $\vec{e_i} = \overrightarrow{uv}$ be the current half-edge of ∂G, for which we aim to construct the zip Z_i and the lines P_i, P'_i (the strip S_i can be easily recovered from Z_i). More precisely, our algorithm will return one half-edge per edge of respective line, so that $\overrightarrow{P_i}$ will be an oriented path starting at u, $\overrightarrow{P'_i}$ will be an oriented path ending at v, while $\overrightarrow{Z_i}$ will keep the half-edges having the origin in P_i and the destination in P'_i (see Algorithm ZIP).

To establish the correctness of this algorithm, it suffices to show that P_i and P'_i induce convex paths of G. Indeed, this would imply that, removing the edges of Z_i, the connected components of the resulting graph are convex sets of G, therefore they are complementary half-planes. Consequently, we will deduce that Z_i is the zip of this pair of half-planes while P_i and P'_i are their lines. First notice that P_i and P'_i are paths because the algorithm alternatively adds half-edges to $\overrightarrow{P_i}$ and $\overrightarrow{P'_i}$. To show for example that the path P_i is convex, by Lemma 1 of [5] it is enough to prove that P_i is locally convex, i.e., if x, y, z are consecutive vertices of P_i, then x and z do not have other common neighbors in G. Suppose not, and let y' be such a common neighbor different from y. Let $\overrightarrow{yx'}$ and $\overrightarrow{zz'}$ be the half-edges which have been added to $\overrightarrow{Z_i}$ at the same iterations of the algorithm at which the half-edges $\overrightarrow{xy}$ and $\overrightarrow{yz}$ have been added to $\overrightarrow{P_i}$. Notice that $\overrightarrow{x'z'}$ is a half-edge of $\overrightarrow{P'_i}$. If x and z are adjacent, then these vertices together with x' and z' induce a 4-cycle, which is impossible. Hence x and z are not adjacent in G. Now, the vertices y and y' must be adjacent, otherwise the vertices x, y, z, y' induce a 4-cycle. If the half-edge $\overrightarrow{yy'}$ belongs

Algorithm ZIP

Input: $\overrightarrow{e_i} \in \partial G$

Output: $\overrightarrow{Z_i}, \overrightarrow{P_i}, \overrightarrow{P'_i}$

$k := 0, \overrightarrow{Z_i} := \{\overrightarrow{e_i}\}, \overrightarrow{P_i} := \emptyset,$

$\overrightarrow{P'_i} := \emptyset, \overrightarrow{e} := twin(\overrightarrow{e_i})$

while $\overrightarrow{e} \notin \partial G$ do

  if k is even

   then

    add $next(\overrightarrow{e})$ to $\overrightarrow{P_i}$ and $prev(\overrightarrow{e})$ to $\overrightarrow{Z_i}$

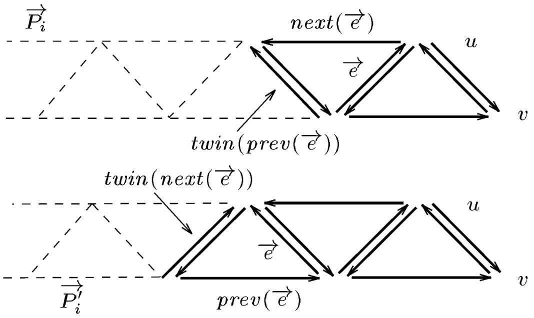
    set $k := k+1$ and $\overrightarrow{e} := twin(prev(\overrightarrow{e}))$

   else

    add $next(\overrightarrow{e})$ to $\overrightarrow{Z_i}$ and $prev(\overrightarrow{e})$ to $\overrightarrow{P'_i}$

    set $k := k+1$ and $\overrightarrow{e} := twin(next(\overrightarrow{e}))$

end do



to $\overrightarrow{Z_i}$, then we will obtain a contradiction with the algorithm, because the half-edges $\overrightarrow{xy}$ and $\overrightarrow{yz}$ will be added to $\overrightarrow{P_i}$ at two consecutive steps of the algorithm. Otherwise, the vertices x, y', z, z', x' will induce a 5-cycle, which is impossible. This shows that the paths P_i and P'_i are indeed locally convex, and therefore convex.

Finally notice that the complexity of this algorithm for a given boundary edge e_i is $O(|Z_i| + |P_i| + |P'_i|) = O(|Z_i|)$. Since every edge of G belongs to exactly two zips, summing over the edges of ∂G, one concludes that the overall complexity of the algorithm is proportional to the number of edges of G, whence it is $O(|V|)$. Analogously, the overall size of the lists Z_i, P_i and P'_i is also linear. Therefore, traversing the paths P_i and P'_i $(i = 1, \ldots, m)$ from the origin to the destination, in total linear time we will compute the weights $\pi(\overrightarrow{P_i}(u)), \pi(\overleftarrow{P_i}(u)), \pi(\overrightarrow{P'_i}(v)), \pi(\overleftarrow{P'_i}(v))$ for all vertices $u \in P_i$ and $v \in P'_i$.

4.3. *Computing the weights of cones and sectors*

Let $C(u; xy)$ be a cone of G bounded by the u-rays $\overrightarrow{P_i}(u)$ and $\overrightarrow{P_j}(u)$ (as we noticed above, one may assume that the cone $C(u; xy)$ is non-degenerated). First observe that the bounding rays of $C(u; xy)$, for example $\overrightarrow{P_i}(u)$, can be constructed in the following way. Start by inserting the half-edge $\overrightarrow{ux}$ in $\overrightarrow{P_i}(u)$ and set $s := x$. At each step, given a current vertex s, turn counterclockwise around s starting from the half-edge next to the half-edge lastly inserted in $\overrightarrow{P_i}(u)$, then leave two edges incident to s and insert in $\overrightarrow{P_i}(u)$ the half-edge $\overrightarrow{ss'}$ of the third edge. Set $s := s'$, and repeat while s is an inner vertex of G. The path $\overrightarrow{P_j}(u)$ is constructed analogously. The single difference is that in the case of $\overrightarrow{P_i}(u)$ the two edges we leave at each iteration will belong to the cone $C(u; xy)$, while in case of $\overrightarrow{P_j}(u)$ they will be outside this cone; see Fig. 4(a).

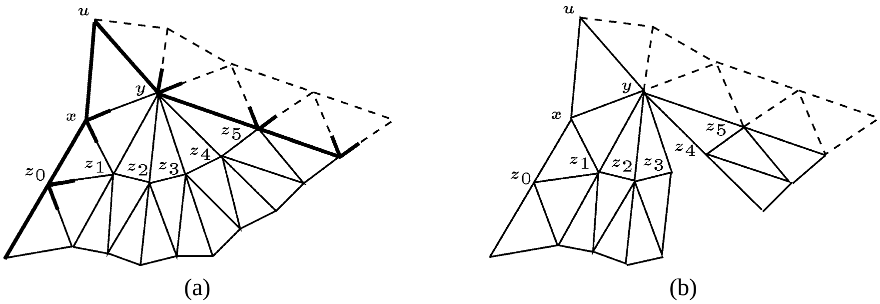


Fig. 4.

Let z_0 be the neighbor of x in $\overrightarrow{P_i}(u)$ (if it exists). Then x may be adjacent to only one other vertex of $C(u;xy) \cup \{y\}$. On the other hand, the vertex y may have several neighbors in $C(u;xy)$ different from u and x, which we denote by $z_1, z_2, \ldots, z_p$. Notice that if z_0 exists, then x has another neighbor in $C(u;xy)$, namely z_1, and in this case z_1 and z_0 are adjacent. Since G is 2-connected, either the vertices $z_0, z_1, \ldots, z_p$ induce a path of G (see Fig. 4(a)) or there exists an index $0 \leqslant k < p$ such that z_k and z_{k+1} are not adjacent and each of $z_0, \ldots, z_k$ and $z_{k+1}, \ldots, z_p$ induces a path of G (see Fig. 4(b)). Special cases occur when z_0 does not exist or $k = p - 1$.

We continue with a formula expressing $\pi(C(u;xy))$ via the weights of the cone $C(x;z_0z_1)$ and of the cones of the form $C(y;z_jz_{j+1})$. For this, first we establish the following equality.

Lemma 3. $C(u;xy) = \{u\} \cup C(x;z_0z_1) \cup (\bigcup_{i=1}^{p-1} C(y;z_iz_{i+1}))$.

Proof. As we noticed above, the half-edges $\overrightarrow{ux}, \overrightarrow{uy}, \overrightarrow{xz_0}, \overrightarrow{xz_1}, \overrightarrow{yz_1}, \ldots, \overrightarrow{yz_p}$ extend in a canonical way to the rays $\overrightarrow{P}(u,x), \overrightarrow{P}(u,y), \overrightarrow{P}(x,z_0), \overrightarrow{P}(x,z_1), \overrightarrow{P}(y,z_1), \ldots, \overrightarrow{P}(y,z_p)$, respectively. Each of these rays is a convex path starting at the origin of the corresponding half-edge and ending at ∂G. Let $s_1, \ldots, s_p \in \partial G$ be the ends of the rays $\overrightarrow{P}(y,z_1), \ldots, \overrightarrow{P}(y,z_p)$ and let t_0 and t_1 be the ends of the rays $\overrightarrow{P}(x,z_0)$ and $\overrightarrow{P}(x,z_1)$. Notice also that $\overrightarrow{P}(x,z_0)$ and $\overrightarrow{P}(y,z_p)$ are subpaths of $\overrightarrow{P}(u,x)$ and $\overrightarrow{P}(u,y)$. Every ray from our list is a bounding ray of one or two consecutive cones occurring in the equality we have to prove. Since the rays are convex, any two rays having a common origin intersect only in this vertex, in particular $\overrightarrow{P}(y,z_i) \cap \overrightarrow{P}(y,z_j) = \{y\}$ holds for all distinct $i, j = 1, \ldots, p$. Next we assert that $\overrightarrow{P}(u,x)$ and $\overrightarrow{P}(y,z_1)$ are disjoint. Suppose the contrary, and pick a vertex t in their intersection. Since $d(x,t) < d(u,t)$, the convexity of $\overrightarrow{P}(u,x)$ implies that $d(x,t) < d(y,t)$. Then x lies on a shortest path between y and t in contradiction with the convexity of $\overrightarrow{P}(y,z_1)$, thus establishing the assertion. Analogously, one can show that the following pairs of rays are disjoint: $\overrightarrow{P}(u,y)$ and $\overrightarrow{P}(y,z_1)$; $\overrightarrow{P}(x,z_1)$ and $\overrightarrow{P}(y,z_2)$; $\overrightarrow{P}(u,x)$ and $\overrightarrow{P}(y,z_i)$ for $i = 2, \ldots, p$.

The cone $C(u;xy)$ consists of the vertices of G located in the region R bounded by the rays $\overrightarrow{P}(u,x)$, $\overrightarrow{P}(u,y)$, and the subpath of ∂G between t_0 and s_p. Analogously, for each $i = 1, \ldots, p-1$, the cone $C(y;z_iz_{i+1})$ consists of the vertices of G located in the region R_i bounded by the rays $\overrightarrow{P}(y,z_i), \overrightarrow{P}(y,z_{i+1})$, and the subpath of ∂G comprised between s_i and s_{i+1}. From what has been proven above follows that the rays $\overrightarrow{P}(y,z_1), \ldots, \overrightarrow{P}(y,z_{p-1})$ partition R into the regions $R'_0, R_1, \ldots, R_{p-1}$ with disjoint interiors; see Fig. 5. Thus every vertex of the cone $C(u;xy)$ located outside R'_0 belongs to some cone $C(y;z_iz_{i+1})$ and, vice versa, the inclusion $\bigcup_{i=1}^{p-1} C(y;z_iz_{i+1}) \subseteq C(u;xy)$ holds. It remains to show that all other vertices of $C(u;xy) - \{u\}$ belong to $C(x;z_0z_1)$ and that $C(x;z_0z_1) \subset C(u;xy)$. Let R_0 be

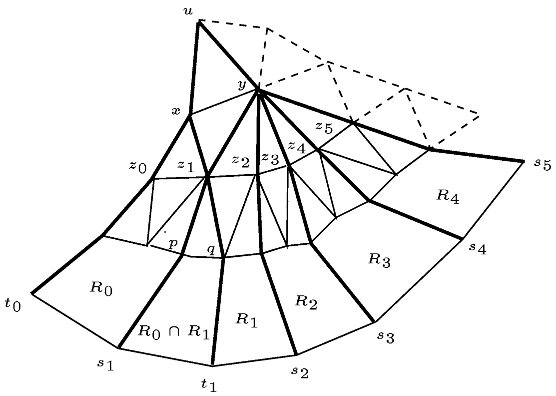


Fig. 5. Proof of Lemma 3.

the region of the plane bounded by the rays $\overrightarrow{P}(x,z_0)$, $\overrightarrow{P}(x,z_1)$, and the subpath of ∂G between t_0 and t_1. Then $R_0 - R_1 \subset R_0'$ and $R_0 \cap V = C(x; z_0z_1)$. Since $R_0' \cap V \subset C(u; xy)$, we deduce that $C(x; z_0z_1) \subset C(u; xy)$. The convexity of the rays $\overrightarrow{P}(x,z_1)$ and $\overrightarrow{P}(y,z_1)$ implies that $\overrightarrow{P}(x,z_1) \cap \overrightarrow{P}(y,z_1) = \{z_1\}$. Therefore $R_0' - R_0$ consists only of the regions bounded by the inner faces (u,x,y) and (x,y,z_1) of G; see Fig. 5 for an illustration. Thus $C(u; xy) - \{u\} \subset C(x; z_0z_1)$, concluding the proof. □

Some cones in the formula from Lemma 3 may overlap. However, two cones whose generators are not incident have only the apex y in common. From the proof of Lemma 3 follows that the intersection of two consecutive non-empty cones $C(y; z_{j-1}z_j)$ and $C(y; z_jz_{j+1})$ is a y-ray. On the other hand, the intersection of the cones $C(x; z_0z_1)$ and $C(y; z_1z_2)$ is again a cone. To show this, notice that the region $R_0 \cap R_1$ is bounded by two subpaths P' and P'' of the convex rays $\overrightarrow{P}(y,z_1)$, $\overrightarrow{P}(x,z_1)$, and the subpath of ∂G between s_1 and t_1. Let p and q be the neighbors of z_1 in those subpaths. Then $P' = \overrightarrow{P}(z_1,p)$ and $P'' = \overrightarrow{P}(z_1,q)$, therefore the vertices of the cone $C(z_1; pq)$ are exactly the vertices of G located in the region $R_0 \cap R_1$, whence the intersection of the cones $C(x; z_0z_1)$ and $C(y; z_1z_2)$ is the cone $C(z_1; pq)$.

From Lemma 3 and previous discussion we obtain the following inclusion-exclusion formula for computing $\pi(C(u; xy))$ (for an illustration of this and subsequent cases see Fig. 6). If z_0 exists and the vertices $z_0, z_1, \ldots, z_p$ induce a path, then

$$\pi\bigl(C(u;xy)\bigr) = \pi(u) + \pi\bigl(C(x;z_0z_1)\bigr) + \sum_{i=1}^{p-1} \pi\bigl(C(y;z_iz_{i+1})\bigr) - \pi\bigl(C(x;z_0z_1) \cap C(y;z_1z_2)\bigr)$$
$$- \sum_{i=2}^{p-1} \pi\bigl(C(y;z_{i-1}z_i) \cap C(y;z_iz_{i+1})\bigr). \quad (1)$$

(Notice that in (1) the weight of y is added $p-1$ times and is subtracted $p-2$ times.) Now, if z_0 does not exist, then simply replace in (1) the cone $C(x; z_0z_1)$ by the degenerate cone $C(x; z_1z_1)$ if x is adjacent to z_1 and by $\{x\}$ otherwise. Finally, if z_0 exists but some consecutive vertices z_k and z_{k+1} are not adjacent, then replace in (1) the cone $C(y; z_kz_{k+1})$ by the degenerate cone $C(y; z_{k+1}z_{k+1})$. In particular, replace $C(y; z_{p-1}z_p)$ by $C(y; z_pz_p)$ if $k = p-1$.

Below we will describe how to organize the computation on G so that each time we wish to compute $\pi(C(u; xy))$, the weights of all cones and rays occurring in the right-hand side of (1) have been already

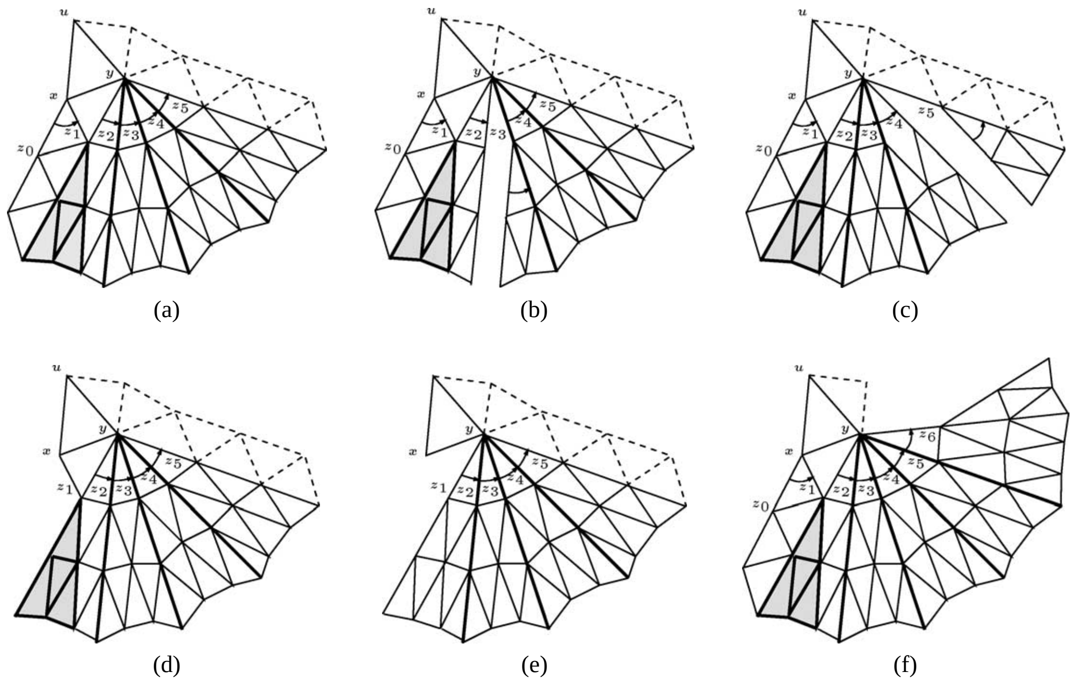


Fig. 6. (a) A non-degenerate case; (b) z_i and z_{i+1} are not adjacent; (c) z_{p-1} and z_p are not adjacent; (d) z_1 is on the boundary; (e) z_1 is on the boundary, x and z_1 are not adjacent; (f) z_p is on the boundary.

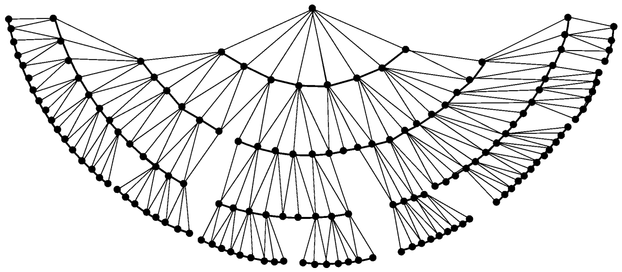


Fig. 7. Levelling of a trigraph.

computed. For this, we pick a vertex c on the outer face of G and perform the levelling of the graph G in the following way: for an integer i define ith *level* L_i to be the subgraph induced by all vertices of G located at distance i from c, see Fig. 7. We call an edge uv of G *horizontal* if both u and v belong to the same level and *vertical* otherwise.

Lemma 4. *Every connected component in each level L_i is a path.*

Proof. Notice that the union of the levels L_j $(j < i)$ is the ball $B_{i-1}(c)$, therefore it is a convex subset of G. Since G is K_4-free, this implies that every vertex of L_i is adjacent to at most two consecutive vertices in the previous level L_{i-1}. In view of (T3), any two adjacent vertices x, y of L_i have a common neighbor u in L_{i-1}. Since G is K_4-free, this common neighbor is necessarily unique.

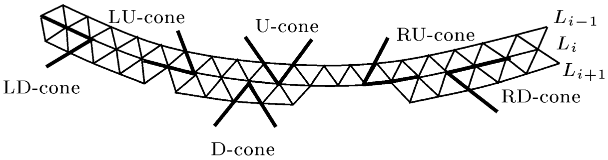


Fig. 8. The classification of cones.

Now, assume by way of contradiction that L_i contains three pairwise adjacent vertices x, y, z. Let z', y', x' be the common neighbors in L_{i-1} of x, y, of x, z, and of y, z, respectively. Convexity of $B_{i-1}(c)$ and the fact that G is K_4-free imply that x', y', z' are distinct and pairwise adjacent. Since x and y have already two neighbors in L_{i-1}, we conclude that the vertices x, y', x', y induce a 4-cycle, which is impossible. Finally, suppose that L_i contains a vertex x adjacent to three other vertices y, z, v. Now, if we consider the common neighbors y', z', v' in L_{i-1} of, respectively, y and x, z and x, and v and x, then each two of them either coincide or are adjacent. If y', z', v' are pairwise distinct, then together with x they will form a K_4, a contradiction with (T2). On the other hand, if all these vertices coincide, then together with x, y, z, v they induce a forbidden $K_{1,1,3}$. Finally, if $y' = z' \neq v'$, then x, y, z, y', v' induce a $K_{1,1,3}$, contrary to (T2). Hence, every vertex of L_i has degree 1 or 2 and L_i does not contain triangles. Thus every connected component of L_i is a path or a cycle. Since the base-point c belongs to the outer face, one can easily see that the second case is impossible, thus L_i consists solely of paths. □

For each path L_i in the levelling consider its first end-vertex in the clockwise traversal of ∂G, refer to it as to the leftmost vertex of L_i and orient L_i from this end-vertex to the right. With respect to the levelling of G, we present the following classification of cones of G. A cone $C(u; xy)$ is called a D-*cone* if $u \in L_{i-1}$ and $x, y \in L_i$, and a RD-*cone* if $u, x \in L_{i-1}$, $y \in L_i$, and x is right from u on L_{i-1}. Analogously one can define the LD-*cones*, the U-*cones*, the LU-*cones* and the RU-*cones*. Call a {D,RD,LD}-cone a *downward cone* and a {U,RU,LU}-cone an *upward cone*. Clearly, every cone of G is of one of these six types; for illustrations see Fig. 8.

The computation of the weights of cones is performed in the following way. First, we sweep G level by level in decreasing order of their distances to c (upward) and compute the weights of downward cones. In order to compute the LD-cones with apices in the ith level, L_i is swept from left to the right, while to compute the analogous RD-cones, L_i is swept from right to the left (the D-cones can be computed at each of these traversals). Since every vertex of L_{i+1} has one or two adjacent neighbors in L_i, one can easily see that every cone used in the computation of the weight of some downward cone with the apex at L_i may occur at most four times at the right-hand side of (1). Therefore the weights of the downwards cones with apices at L_i can be computed in time proportional to the number of edges in the subgraph induced by $L_i \cup L_{i+1}$, whence the overall computation of weights of downward cones is linear.

Now, to compute the weights of upward cones, we sweep the levels of the graph G in increasing order of their distances to c (downward). At stage i, we traverse the level L_i from right to left, and for every vertex $y \in L_i$, we compute the weights of all upward cones having y as the left end-vertex of their generator, i.e., of all cones $C(u; xy)$ such that the half-edge $\overrightarrow{xy}$ occurs in the counterclockwise traversal of the inner face (u, x, y). However, computing $\pi(C(u; xy))$ directly via (1) would not yield a linear-time algorithm because every cone with apex y appears in the right-hand side of this formula for all upward cones $C(u; xy)$ except a constant number. Instead, we proceed in the following way. Let $z_0, z_1, \ldots, z_{p-1}, z_p$ be the neighbors of y in G ordered in the counterclockwise order, where

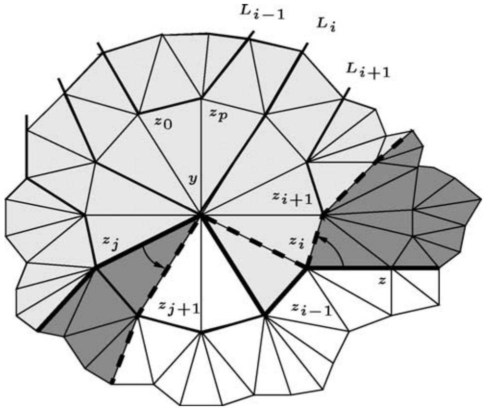


Fig. 9. $C(z_i; z_{i+1}y)\Delta C(z_{i-1}; z_i y) = C(y; z_j z_{j+1}) \cup C(z_i; zz_{i+1})$.

$z_0, z_p \in L_{i-1}, z_1, z_{p-1} \in L_i$ and the remaining neighbors are in L_{i+1}. First, applying (1) we compute the weight of the rightmost upward cone $C(z_{p-1}; z_p y)$. Then we successively update this weight by turning around the vertex y in clockwise order. Assume for example that we wish to compute the weights of the upward cones $C(z_{i-1}; z_i y)$ ($i = 2, \ldots, p-1$). For this, notice that the symmetric difference between two consecutive cones $C(z_i; z_{i+1}y)$ and $C(z_{i-1}; z_i y)$ consists of two cones $C(y; z_j z_{j+1})$ and $C(z_i; zz_{i+1})$, where z is the common neighbor of z_i and z_{i+1} different from y (if z does not exist, then the second cone is the degenerated cone $C(z_i; z_{i+1}z_{i+1})$). As we will establish below, one can suppose that the weights of these two cones have been already computed. Now, knowing $\pi(C(z_{i-1}; z_i y))$, the weight of the cone $C(z_i; z_{i+1}y)$ is obtained by setting

$$\pi\big(C(z_i; z_{i+1}y)\big) := \pi\big(C(z_{i-1}; z_i y)\big) - \pi\big(C(z_i; zz_{i+1})\big) + \pi\big(C(y; z_j z_{j+1})\big);$$

since the last two cones occurring in this formula are downward cones, their weights are already known (for an illustration see Fig. 9). Clearly, the complexity of performing these computations for a given vertex y is proportional to its degree, therefore the overall time of computing the upward cones is also linear.

The correctness of this algorithm follows from the following result.

Lemma 5. *If $C(u; xy)$ is the current cone, then the weights of cones arising at the right-hand side of* (1) *have been already computed.*

Proof. The basic ingredients of the proof are Lemma 4 and the following facts about $C(u; xy)$. First, from Lemma 2 we conclude that the cone $C(u; xy)$ is convex and that its rays are convex paths. Second, for every vertex $z \neq u$ of $C(u; xy)$ every shortest path between u and z intersects the generator $\{x, y\}$. As above, by $z_0, z_1, \ldots, z_p$ we denote the neighbors of y and/or x in $C(u; xy)$. From previous properties of cones, we conclude that neither of these vertices is adjacent to u. Now, suppose that the levelling of G has n levels and that u belongs to L_i. We proceed by induction on $n - i$ for downward cones and by induction on i for upward cones.

Case 1. $C(u; xy)$ is a downward cone.

If $x, y \in L_{i+1}$ (i.e., $C(u; xy)$ is a D-cone), however some z_j belongs to L_i, then z_j and u must be adjacent because they have a common neighbor outside the ball $B_i(c)$, which is impossible. So, assume without loss of generality that $x \in L_{i+1}$ and $y \in L_i$. Since u is not adjacent to z_0, we conclude that $z_0, z_1 \notin B_i(c)$, thus the weight $\pi(C(x; z_0z_1))$ is already known in view of induction hypothesis. As to the cones $C(y, z_jz_{j+1})$, $i = 1, \ldots, p-1$, assume by way of contradiction that some z_j belongs to the level L_{i-1}. Since z_j is not adjacent to u and $u, y \in L_i$, by (T3) there exists a common neighbor $z \neq z_j$ of u and y one step closer to c. Since $z, z_j \in L_{j-1}$ and both these vertices are adjacent to y, the convexity of the ball $B_{i-1}(c)$ yields that z is adjacent to z_j. Then $z \in I(u, z_j) \subset C(u; xy)$, which is impossible.

Case 2. $C(u; xy)$ is an upward cone.

First assume that $x, y \in L_{i-1}$, and pick a cone from the right-hand side of (1), say the cone $C(y; z_jz_{j+1})$. If the vertices of its generator belong to the levels L_{i-1} and L_{i-2}, then, by the induction hypothesis, the weight of this cone is known. On the other hand, if one vertex of its generator belongs to L_{i+1} and another one to L_i or L_{i+1}, then $C(y; z_jz_{j+1})$ is a downward cone, therefore its weight has been computed at previous stage. The case when $C(u; xy)$ is a LU- or RU-cone is analogous subject to minor modifications. For example, if, say $y \in L_i$ and $x \in L_{i-1}$, then no cone $C(y; z_jz_{j+1})$ may have both z_j and z_{j+1} in L_{i-1}: the convexity of $B_{i-1}(c)$ then implies that x, y, z_j, z_{j+1} are pairwise adjacent and we get a forbidden K_4. In all other cases, $C(y; z_jz_{j+1})$ is either a downward or an upward cone whose weight has been already computed due to induction hypothesis. □

4.4. Computing the weights of half-planes

Let H_i, H_i' be a pair of complementary half-planes defined by the zone Z_i. Let uv and $u'v'$ be the boundary edges of Z_i so that u, v' are the end-vertices of P_i and v, u' are the end-vertices of P_i'. We will show how to compute $\pi(H_i)$ ($\pi(H_i')$ can be computed analogously but using the vertex u'). Denote by $z_1 := v, \ldots, z_p$ the neighbors of the vertex u ordered clockwise. Then $z_2, \ldots, z_p$ are the neighbors of u lying in the half-plane H_i; see Fig. 10. Now notice that H_i is the union of the non-degenerated

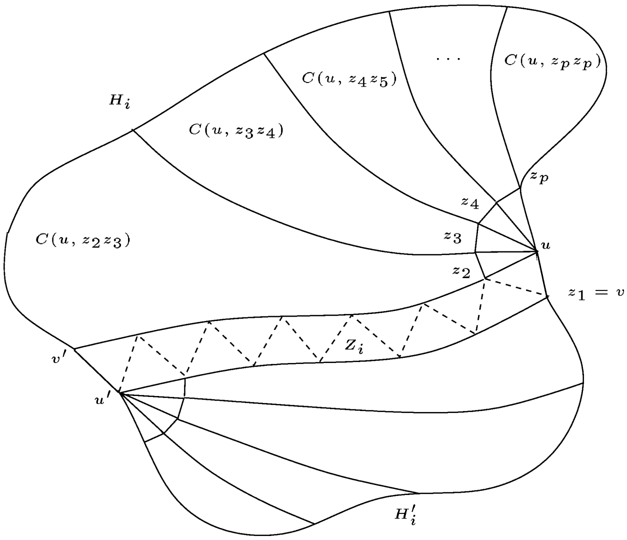


Fig. 10.

cones $C(u; z_j z_{j+1})$ $(j = 2, \ldots, p-1)$ and of the degenerated cone $C(u; z_p z_p)$. The u-rays defined by the edges $uz_3, \ldots, uz_{p-1}$, being the intersection of two consecutive cones, are counted twice. Hence

$$\pi(H_i) = \sum_{j=2}^{p} \pi\big(C(u; z_j z_{j+1})\big) - \sum_{j=2}^{p-1} \pi\big(C(u; z_j z_{j+1}) \cap C(u; z_{j+1} z_{j+2})\big),$$

where $C(u; z_p z_{p+1})$ stands for the degenerated cone $C(u; z_p z_p)$. This shows that the weight of H_i can be computed in time proportional to the degree of the vertex u. The vertices u and v are separated by two pairs of complementary half-planes, therefore u will be involved in computing the weights of two half-planes only. This proves that the weights of the half-planes of G can be computed in time proportional to the sum of degrees of vertices of ∂G, i.e., in linear time.

4.5. The algorithm MEDIAN SET and its complexity

Summarizing the discussion from previous subsections, we outline the following algorithm for computing the median set $\mathrm{Med}(\pi)$ of a trigraph G.

Algorithm MEDIAN SET
Input: A trigraph G in the form of a doubly-connected edge list and a weight function π
Output: The median set $\mathrm{Med}(\pi)$
1. Compute the zips Z_i, their strips S_i, and the lines P_i, P_i' $(i = 1, \ldots, m)$;
2. For $i = 1, \ldots, m$ and all vertices $u \in P_i, v \in P_i'$ compute the weights $\pi(\overrightarrow{P_i}(u))$, $\pi(\overleftarrow{P_i}(u))$, $\pi(\overrightarrow{P_i'}(v))$, $\pi(\overleftarrow{P_i'}(v))$ of the u- and v-rays;
3. Compute the weights $\pi(C(u; xy))$ of the cones $C(u; xy)$ of G, and then compute the weights $\pi(S(uv; x))$ of the sectors $S(uv; x)$ of G;
4. Compute the weights $\pi(H_i)$ and $\pi(H_i')$ of the half-planes H_i and H_i' $(i = 1, \ldots, m)$;
5. For each edge uv of G compute $\pi(W(u, v))$ and $\pi(W(v, u))$ as the difference between weights of a half-plane and a sector computed in steps 3 and 4;
6. Construct the graph $\overrightarrow{G_\pi}$;
7. Return the set $\mathrm{Med}(\pi)$ consisting of all vertices of G having no outgoing edges in $\overrightarrow{G_\pi}$.

While describing in details steps 1–4 of the algorithm, we established that the complexity of each of these steps is linear. When the weights of complementary half-planes H_i, H_i' are computed, then they are broadcasted to all edges of the zip Z_i. Now, given an edge $uv \in Z_i$, the weights of $W(u, v)$ and $W(v, u)$ can be found in constant time as noticed in step 5 and illustrated in Fig. 3. Hence step 5 needs $O(|E|)$ operations, the same order as the steps 6 and 7. Since $|E| \leqslant 3|V| - 2$ because G is planar, we conclude that the complexity of the algorithm MEDIAN SET is $O(|V|)$. This algorithm can be modified (even simplified) in order to compute the median sets of squaregraphs (we skip the straightforward details). Concluding, we obtain the following result.

Theorem 1. *For every weight function π defined on vertices of a trigraph or a squaregraph $G = (V, E)$, the median set* $\mathrm{Med}(\pi)$ *can be computed in linear time* $O(|V|)$.

Acknowledgements

We would like to acknowledge the referees for several useful suggestions.

References

[1] S. Alstrup, J. Holm, M. Thorup, Maintaining center and median in dynamic trees, in: 7th Scandinavian Workshop on Algorithm Theory, SWAT'00, 2000, pp. 46–56.
[2] G. Antonoiu, P.K. Srimani, A self-stabilizing distributed algorithm to find the median of a tree graph, J. Computer System Sci. 58 (1) (1999) 215–221.
[3] V. Auletta, D. Parente, G. Persiano, Dynamic and static algorithms for optimal placement of resources in a tree, Theoret. Comput. Sci. 165 (1996) 441–461.
[4] H.-J. Bandelt, J.P. Barthélémy, Medians in median graphs, Discrete Appl. Math. 8 (1984) 131–142.
[5] H.-J. Bandelt, V. Chepoi, Decomposition and l_1-embedding of weakly median graphs, European J. Combin. 21 (2000) 701–714.
[6] H.-J. Bandelt, V. Chepoi, Graphs with connected medians, SIAM J. Discrete Math. 15 (2002) 268–282.
[7] E. Barcucci, A. Del Lungo, M. Nivat, R. Pinzani, Medians of polyominoes: a property for the reconstruction, Internat. J. Imag. Syst. Technol. 8 (1998) 69–77.
[8] J.P. Barthélemy, B. Monjardet, The median procedure in cluster analysis and social choice theory, Math. Social Sci. 1 (1981) 235–268.
[9] O. Baues, N. Peyerimhoff, Curvature and geometry of tessellating plane graphs, Discrete Comput. Geom. 25 (2001) 141–159.
[10] M. Bernstein, V. de Silva, J.C. Langford, J.B. Tenenbaum, Graph approximations to geodesics on embedded manifolds, manuscript, 2000, 24 pp.
[11] P. Bose, A. Maheshwari, P. Morin, Fast approximations for sums of distances, clustering and the Fermat–Weber problem, Computational Geometry 24 (2003) 135–146.
[12] M.R. Bridson, A. Haefliger, Metric Spaces of Non-Positive Curvature, Springer, Berlin, 1999.
[13] S.C. Bruell, S. Ghosh, M.H. Karaata, S.V. Pemmaraju, Self-stabilizing algorithms for finding centers and medians of trees, SIAM J. Comput. 29 (1999) 600–614.
[14] V. Chepoi, Graphs of some CAT(0) complexes, Adv. Appl. Math. 24 (2000) 125–179.
[15] V. Chepoi, F. Dragan, Computing the median point of a simple rectilinear polygon, Inform. Process. Lett. 49 (1994) 281–285.
[16] V. Chepoi, F. Dragan, Y. Vaxès, Center and diameter problems in plane triangulations and quadrangulations, in: 13th ACM-SIAM Annual Symposium on Discrete Algorithms, SODA'02, 2002, pp. 346–355.
[17] A. Daurat, A. Del Lungo, M. Nivat, The medians of discrete sets according to some linear metrics, Discrete Comput. Geom. 23 (2000) 465–483.
[18] M. de Berg, M. van Kreveld, M. Overmars, O. Schwarzkopf, Computational Geometry. Algorithms and Applications, Springer, Berlin, 1997.
[19] A. Del Lungo, M. Nivat, R. Pinzani, L. Sorry, The medians of discrete sets, Inform. Process. Lett. 65 (1998) 293–299.
[20] A.J. Goldman, Optimal center location in simple networks, Transport. Sci. 5 (1971) 212–221.
[21] A.J. Goldman, C.J. Witzgall, A localization theorem for optimal facility placement, Transport. Sci. 4 (1970) 406–409.
[22] Y. Métivier, N. Saheb, Medians and centers of polyominos, Inform. Process. Lett. 57 (1996) 175–181.
[23] F.R. McMorris, R.C. Powers, The median procedure in a formal theory of consensus, SIAM J. Discrete Math. 8 (1995) 507–516.
[24] P. Soltan, V. Chepoi, The solution of the Weber problem for discrete median metric spaces, Trudy Tbilisskogo Mat. Inst. 85 (1987) 52–76 (in Russian).
[25] B.C. Tansel, R.L. Francis, T.J. Lowe, Location on networks: a survey I, II, Management Sci. 29 (1983) 482–511.
[26] J.B. Tenenbaum, V. de Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (2000) 2319–2323.
[27] M.L.J. van de Vel, Theory of Convex Structures, North-Holland, Amsterdam, 1993.
[28] B. Zelinka, Medians and peripherians of trees, Arch. Math. 4 (1968) 87–95.