Karlsruher Institut für Technologie

Karlsruhe Reports in Informatics 2012,19
Edited by Karlsruhe Institute of Technology,
Faculty of Informatics
ISSN 2190-4782

On partitioning a plane graph by plane curves

Roland Glantz, Henning Meyerhenke

2012

KIT – University of the State of Baden-Wuerttemberg and National Research Center of the Helmholtz Association

Fakultät für **Informatik**

On partitioning a plane graph by plane curves

ROLAND GLANTZ*, HENNING MEYERHENKE†

Institute of Theoretical Informatics, Karlsruhe Institute of Technology (KIT), Box 6980, D-76128 Karlsruhe, Germany

October 4, 2012

Abstract

Traditional geometric methods for partitioning an embedded graph neglect the connectivity of the vertices via edges. For a special class of graphs, the so-called ℓ_1 graphs, however, one can devise geometric graph partitioning methods that incorporate the edges. A plane graph G is an ℓ_1 graph if and only if it an isometric subgraph of a half-cube, a bipartite half of a hypercube. Such a graph comes with vertex labels which encode all its convex cuts, *i. e.,* cuts which yield subgraphs that are closed with respect to shortest paths. Convex cuts are useful for graph partitioning as they induce well-shaped partitions. The vertex labels also encode distances between any pair of vertices and thus provide guidance for finding shortest paths in the subgraph.

In this paper we first generate a "well-arranged" subgraph G_w of G. This means that G_w comes with an arrangement of embedded paths running across edges and through faces of G_w such that (i) any edge of G_w is hit by exactly two paths, (ii) the paths do not intersect themselves, and (iii) any two paths cross each other at most once. We prove that G_w is an isometric subgraph of a half-cube. In particular, the arrangement of G_w generates all convex cuts of G_w.

To obtain a partition of G, we extend the paths of the arrangement towards the unbounded face of G. The extensions are guided by a gradient vector field on G^* (the dual of G) and indicate steepest descent paths with respect to the distance to the vertex representing the unbounded face of G. The extended paths are free of self-intersections and each pair of paths intersects at most once. Our algorithm for computing all extended paths runs in linear time with respect to the maximum of $|E(G)|$ and the number of edges in all extended paths.

Thus, combining the generation of G_w with the extension of the paths constitutes a geometric method for partitioning a plane graph G that does not rely on coordinates. The density of the extended paths (any face of G is intersected by at least two paths) suggests to use them as a basis for methods solving more complex partitioning problems, *e. g.,* with constraints on the sizes of the subgraphs.

1 Introduction

A common variant of the graph partitioning (GP) problem asks for the division of a graph's vertex set into (approximately) equally sized subsets such that the size [or weight, respectively] of the *cutset*, i. e., the set of edges with endpoints in different subsets, is minimized. Despite advances in approximation [14, 12] and exact algorithms [4] for this $\mathcal{NP}$-hard problem and similar ones, heuristics are dominant in practice. A class of popular global heuristics are geometric methods which partition the graph's vertex set by hyperplanes. Unfortunately, such geometric GP methods are limited by their inability to account for the connectivity of G. Thus, the number of edges that are cut is often high compared to (more complex and more time consuming) methods that take the connectivity of G into account [13].

In this paper we consider plane graphs and compute a collection of embedded paths that partition G. We expect this collection to be useful as a basis for methods solving more complex partitioning problems such as the one above.

The rationale behind our methods is as follows. If G were a so-called ℓ_1 graph [3], it could be embedded into $\mathbb{R}^n$ (usually $n \gg 2$) such that the distance between vertices u and v of G (*i. e.,* the number of edges on a shortest path from u to v) equals the ℓ_1 distance of the embedded vertices in $\mathbb{R}^n$. (The ℓ_1 distance between $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n)$ is $\sum_{i=1}^{n} |x_i - y_i|$.) In this case one can devise geometric GP methods in $\mathbb{R}^n$ that incorporate the connectivity of G. Specifically, (convex) partitions of $\mathbb{R}^n$ by hyperplanes orthogonal to the unit vectors of $\mathbb{R}^n$ yield a plethora of convex cuts of G. For the role of convex cuts in graph partitioning see Section 1.1.

*Electronic address: roland.glantz@kit.edu
†Electronic address: meyerhenke@kit.edu

Planar ℓ_1 graphs are tightly connected to hypercubes since every planar ℓ_1 graph G has a so-called *scale two embedding* into a hypercube H [3]. This means that G can be embedded into a hypercube H such that the distance between u and v in G equals half the distance between u and v in H for all u, v in G. Equivalently, G has an isometric embedding, *i. e.,* a scale one embedding, into a half-cube $\frac{H}{2}$ of H (a half-cube is a bipartite half of a hypercube).

A hypercube in $\mathbb{R}^n$ can be partitioned into two halves (these are not half-cubes) by a hyperplane orthogonal to a unit vector of $\mathbb{R}^n$. The arrangement of these n hyperplanes partitions G such that any edge of G is intersected by two hyperplanes. Indeed, an edge of G is an edge of $\frac{H}{2}$ which, in turn, corresponds to a path of length two in H and is thus intersected by two hyperplanes. Moreover, for any pair of hyperplanes there exists at most one edge of G that is intersected by both hyperplanes. Thus the arrangement of hyperplanes gives rise to an arrangement of plane curves in $\mathbb{R}^2$ that partition G. For an example see the colored "curves" in Figure 1b. The arrangement becomes a plane graph A if we put vertices at the ends of the curves and at points of intersection between curves. See the red vertices in Figure 1b. The plane curves thus turn into a collection $\mathcal{E}(G)$ of embedded paths of A such that (1) none of the paths intersects itself, (2) any pair of distinct paths intersects at most once and (3) any edge of G is intersected by exactly two paths.

For a plane graph G that is not an ℓ_1 graph, we relax the requirements on $\mathcal{E}(G)$.

Property 1.1 (Collection $\mathcal{E}'(G)$ of embedded paths)
The collection $\mathcal{E}'(G)$ for partitioning G has to meet the following criteria.

1. *Any path in $\mathcal{E}'(G)$ must not intersect itself,*
2. *any pair of distinct paths in $\mathcal{E}'(G)$ must not intersect more than once,*
3. *any face of G is intersected by at least two paths in $\mathcal{E}'(G)$, and*
4. *any path in $\mathcal{E}'(G)$ intersects an edge of G that is intersected by exactly one other path in $\mathcal{E}'(G)$.*

The purpose of Item (4) is to exclude collections of paths which have many (parallel) paths that yield identical partitions of G.

1.1 Related work.

The fundamental notion of convexity can be used to draw a connection between continuous objects in a metric space and discrete objects like graphs. A subgraph S of a graph G is *convex* if for all $u, v \in V(S)$, all shortest paths between u and v are contained in S. Following Artigas et al. [1], a convex k-partition in a graph is a partition of the vertex set into k convex sets. If G has a convex k-partition, then G is said to be k*-convex*. Deciding whether a graph is k-convex, is $\mathcal{NP}$-complete for a fixed $k \geq 2$ [1].

Graph partitions with particular properties are of high interest in many applications. Among the practical ones are parallel computing [13] and VLSI design [7]. Sample applications benefiting from the convexity property of a cut are parallel numerical simulations using certain iterative linear solvers. For some solvers used in these simulations, the *shape* of the partitions, in particular short boundaries, small aspect ratios, but also connectedness and smooth boundaries, plays a significant role [10]. Convex cuts typically admit these properties. Another example is the preprocessing of road networks for shortest path queries by partitioning according to natural cuts [5]. The definition of a natural cut is not as strict as that of a convex cut, but they have a related motivation.

The classical planar separator theorem [9] can be used to design divide-and-conquer algorithms for planar graphs. In such algorithms the vertex set is recursively partitioned into subsets of respective size not larger than $\frac{2}{3}|V|$. The solutions of the recursive subproblems are combined at their interface, whose size is $\mathcal{O}(\sqrt{|V|})$ by the theorem.

Our approach to finding collections of paths that have Property 1.1 is motivated by the connections between convex cuts and alternating cuts as described in [3]. As an example we would like to mention Proposition 2 in [3]. It states that a cut of a plane G is alternating if and only if it is convex, provided that G has a certain property. Although the property is not used in this paper, Proposition 2 and others provide crucial insights for the methods we propose here. The subgraph G_w of G that we define in Section 3.3 is a plane graph whose alternating cuts coincide with its convex cuts. The alternating cuts in [3] give rise to the alternating paths in this paper.

1.2 Outline and contribution.

In Section 2 we specify the class of plane graphs G which serve as input for our method. The main purpose of Section 3 is to represent the collection of alternating cuts of G, as defined in [3], as a collection $\mathcal{E}(G)$ of alternating paths embedded in $\mathbb{R}^2$. In particular, any edge of G is intersected by exactly two paths from $\mathcal{E}(G)$, and the paths are straight from a local perspective. This means that a path enters and leaves a face F of G through edges that are opposite edges of F (see the colored paths in Figure 1b). In Section 3 we also formulate conditions on $\mathcal{E}(G)$, i.e., that no path in $\mathcal{E}(G)$ intersects itself and that two paths in $\mathcal{E}(G)$ intersect at most once. The "well-arranged" graphs below are the graphs whose collections meet these conditions.

Recall that our aim is to find a collection $\mathcal{E}'(G)$ of embedded alternating paths which partition G such that the collection has Property 1.1. To this end, our search for $\mathcal{E}'(G)$ is centered around a well-arranged subgraph G_w of G. In Section 4 we show that any well arranged subgraph is an isometric subgraph of a half-cube. Thus G_w can be partitioned naturally into convex subgraphs.

In Section 5 we first present a linear-time method for finding G_w. In order to arrive at a collection $\mathcal{E}'(G)$ that partitions G and that has Property 1.1, we extend the paths in $\mathcal{E}(G_w)$ towards the unbounded face. Specifically, the extensions are guided by a gradient vector field on the dual of G that indicates steepest descent paths with respect to the distance to the vertex representing the unbounded face of G (this gradient vector field has already been used to find G_w). The running time of transforming $\mathcal{E}(G_w)$ into $\mathcal{E}'(G)$ is linear in the total number of edges of paths in $\mathcal{E}'(G)$. The running time of the rest of our method amounts to $\mathcal{O}(|E(G)|)$.

The collection $\mathcal{E}'(G)$ arises from the combination of two geometric partitioning concepts, none of which depends on locations of vertices or lengths of edges. The first geometric concept is partitioning through arrangements of hyperplanes [2]. In fact, G_w is an isometric subgraph of a half-cube and the convex subgraphs cut out by the paths in $\mathcal{E}(G_w)$ correspond to half-spaces cut out of $\mathbb{R}^n$ by hyperplanes. Here n is the dimension of the half-cube into which G_w is embedded isometrically. The second geometric concept is partitioning through paths of steepest descent. We use this concept to find G_w and to transform $\mathcal{E}(G_w)$ into $\mathcal{E}'(G)$. Practically the latter concept helps to keep the cutsets of a partition small. To the best of our knowledge, the combination of geometric concepts just described is new in the realm of graph partitioning methods.

2 Preliminaries

All primal graphs and their geometric duals considered in this paper are finite, undirected, connected, free of self-loops and plane. If G is such a graph, we write $G = G(V, E)$, where V is the set of G's vertices and E is the set of G's edges. Since G is plane, we may identify V with a set of points in $\mathbb{R}^2$ and E with a set of plane curves that intersect only at their end points, which, in turn, make up V. For $e \in E$ with end points u, v ($u \neq v$) we sometimes abuse notation by writing $e = \{u, v\}$, being aware of the fact that, due to parallel edges, e is not necessarily determined by u and v. We denote the standard metric on G by $d_G(\cdot, \cdot)$. In this metric the distance between $u, v \in V$ amounts to the number of edges in a shortest path from u to v.

We denote the geometric dual of a primal graph $G = (V, E)$ by $G^* = (V^*, E^*)$. For $E_0 \subseteq E$, let E_0^* denote the set of edges in E^* that are dual to the edges in E_0. For $e \in E$, e^* is the edge of E^* that is dual to e. If F is a face of G, we write $E(F)$ for the set of edges that bound F. Finally, F_∞ denotes the unbounded face of G, and u_∞^* denotes the vertex of G^* that represents F_∞.

3 Well-arranged plane graphs

The main purpose of this section is to introduce well-arranged plane graphs. To this end, we define a multiset of non-embedded alternating paths for any plane graph G in Section 3.1. After that, Section 3.2 is devoted to embedding these paths. This will eventually give rise to arrangements of alternating paths and well-arranged plane graphs in Section 3.3. There we also define a weak dual of an arrangement, which will turn out to be a partial cube in Section 4.

3.1 Alternating paths

Intuitively, an embedded alternating path P runs through a face F of G such that the edges through which P enters and leaves F are opposite—or nearly opposite because, if $|E(F)|$ is odd, there is no opposite edge, and P has to make a slight turn to the left or to the right. The exact definitions for not yet embedded alternating paths are as follows.

Definition 3.1 (Opposite edges, left, right, unique opposite edge)
Let $F \neq F_\infty$ be a face of G, and let $e, f \in E(F)$. Then e and f are called opposite edges *of F if the lengths of the two paths induced by $E(F) \setminus \{e, f\}$ differ by at most one. If the two paths have different lengths, f is called the* left [right] opposite edge *of e if starting on e and running clockwise around F, the shorter [longer] path comes first. Otherwise, e and f are called* unique opposite edges.

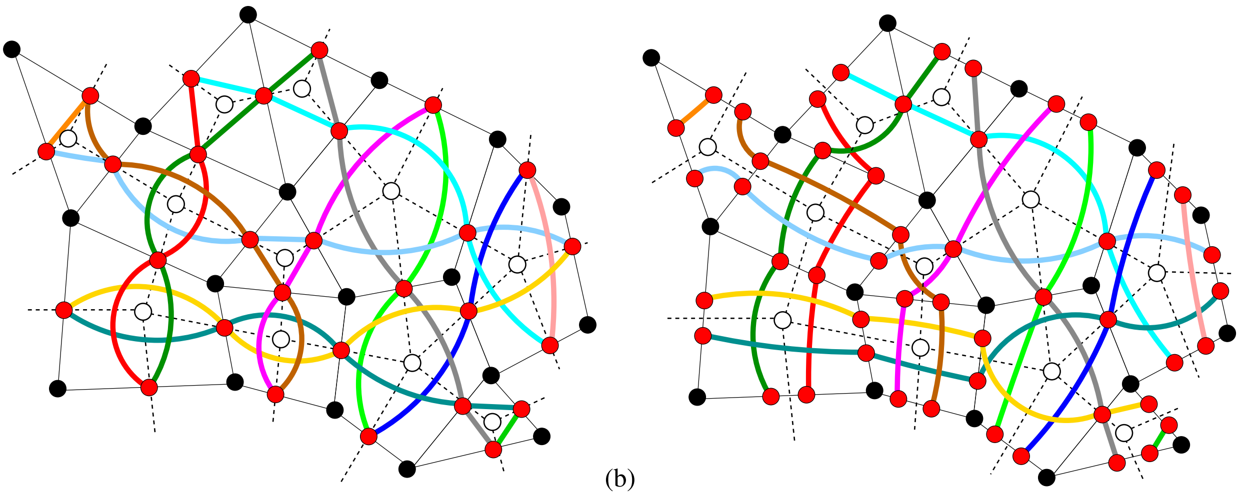
Definition 3.2 (Alternating path graph $A(G) = (V_A, E_A)$, edge through face F of G)
The alternating path graph *$A(G) = (V_A, E_A)$ of $G = (V, E)$ is the (non-plane) graph with $V_A = E$ and E_A consisting of all two-element subsets $\{e, f\}$ of E such that e and f are opposite edges of some face $F \neq F_\infty$. Such an edge $\{e, f\}$ is sometimes referred to as* edge through F.

The alternating path graph defined above only provides the edges for the multiset of alternating paths defined next. We have to resort to a *multiset* of alternating paths (with multiplicities one and two) because, as we will see in Section 3.2, a single non-embedded path may give rise to two embedded alternating paths.

Definition 3.3 ((Multiset $\mathcal{P}(G)$ of) Alternating paths in $A(G)$)
A maximal path $P = (v_A^1, e_A^1, v_A^2, \ldots e_A^{n-1}, v_A^n)$ in $A(G) = (V_A, E_A)$ is called alternating *if*

- *v_A^i and v_A^{i+1} are opposite for all $1 \leq i \leq n-1$ and*
- *if v_A^{i+1} is the left [right] opposite of v_A^i, and if j is the minimal index greater than i such that v_A^j and v_A^{j+1} are not straight opposites (and j exists at all), then v_A^{j+1} is the right [left] opposite of v_A^j.*

The multiset $\mathcal{P}(G)$ contains all alternating paths in $A(G)$: the multiplicity of P in $\mathcal{P}(G)$ is two if v_A^{i+1} is a straight opposite of v_A^i for all $1 \leq i \leq n-1$, and one otherwise.



(a) (b)

Figure 1: Primal graph: Black vertices, thin solid edges. Dual graph: White vertices, dashed edges. (a) Multiset $\mathcal{P}(G)$ of alternating paths: Red vertices, thick solid lines. The paths in $\mathcal{P}(G)$ are colored. In this ad-hoc drawing all alternating paths that contain a vertex v_A (edge e of G) go through the same point on e, *i. e.,* where a red vertex was placed. (b) Collection $\mathcal{E}(G)$ of embedded alternating paths: Red vertices, thick solid colored lines.

3.2 Constructing a plane embedding of alternating paths

In this section we demonstrate how we derive a collection of embedded alternating paths $\mathcal{E}(G)$ from a multiset $\mathcal{P}(G)$ of (non-embedded) alternating paths. A path in $\mathcal{P}(G)$ with multiplicity $m \in \{1, 2\}$ will give rise to m embedded paths in $\mathcal{E}(G)$. Visually, we go from Figure 1a to Figure 1b.

Notation 3.4 *The collection of embedded alternating paths is denoted by $\mathcal{E}(G)$.*

The following procedure for embedding alternating paths is guided by the objective to minimize intersections. We specify only the essential characteristics of the embedding, *e. g.,* whether the two alternating paths entering a face F of G via an edge

$e \in E(F)$ do so by going through one or two points on e (the two paths cross on e if and only if it is one point). The exact position of the point(s) on e is irrelevant for the rest of the paper, as is the exact course of an alternating path through F. For any path P through F it will always be clear, however, which other paths through F are intersected by P in $F \cup E(F)$ and where the intersection takes place (on which edge of $E(F)$ or in F). We call a face of F even [odd] if $|E(F)|$ is even [odd].

If $F \neq F_\infty$ is an even face of G, and if e, f are unique opposite edges in $E(F)$, then there exist $P_1, P_2 \in \mathcal{P}(G)$ with parallel edges e_1 and e_2 from e to f such that e_1 and e_2 are edges through F in the sense of Definition 3.2. For an illustration see Figure 2a. In $\mathcal{E}(G)$ the edge e_1 [e_2] is a non-self-intersecting plane curve from a point p_1^e on the interior of e to a point p_1^f on the interior of f [p_2^e on the interior of e to a point p_2^f on the interior of f]. We embed the edges e_1 and e_2 such that $p_1^e \neq p_2^e$, $p_1^f \neq p_2^f$, and $e_1 \cap e_2 = \emptyset$ (see again Figure 2a).

Now let F be an odd face of G, and let $e \in E(F)$. If e bounds more than one bounded face and if the other face is even, the two alternating paths through e must intersect with the interior of e at two points $p_1^e \neq p_2^e$ (see above). If the other face is odd, too, we let the two alternating paths through e intersect at a point in the interior of e (see Figure 2b). We embed the edges through (odd) F such that two edges intersect in the interior of F if and only if they enter F through different faces. Two edges that intersect in F's interior must do so only once, and we prohibit intersections of more than two edges at a single point.

If e is an edge of G that bounds F_∞ and another face F, we embed the alternating paths ending at e such that they end in different points in the interior of e (see Figure 1b).

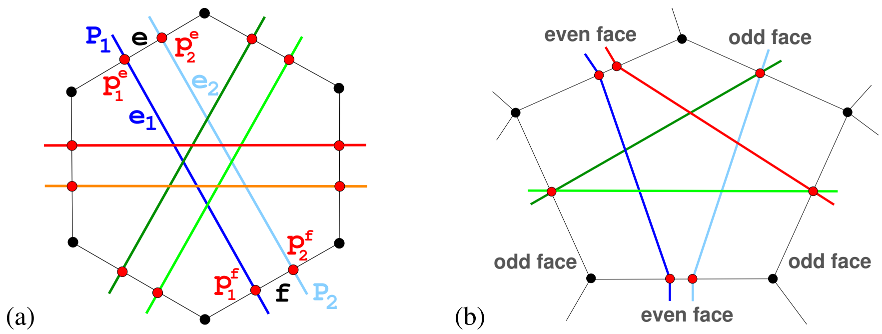


Figure 2: Intersection pattern of (a) a hexagon and (b) a pentagon.

3.3 Well-arranged plane graphs and their weak duals

In this section we first specify a special class of collections of alternating paths, *i. e., arrangements* of alternating paths. In the following sections we will see that plane graphs whose collection of alternating paths falls into this class, *i. e.,* the well-arranged plane graphs defined below, are graphs that have natural partitions into convex subgraphs.

Definition 3.5 (Arrangement of alternating paths, well-arranged graph G_w)
$\mathcal{E}(G)$ is called an arrangement of alternating paths *if*

1. *none of the alternating paths is a cycle,*
2. *none of the alternating paths intersects itself, and*
3. *there exist no paths $P_1 \neq P_2 \in \mathcal{E}(G)$ such that $P_1 \cap P_2$ contains more than one point.*

A plane graph G_w is called well-arranged *if $\mathcal{E}(G_w)$ is an arrangement of alternating paths.*

The notion of an arrangement of alternating paths can be seen as a generalization of the notion of an arrangement of pseudolines [2]. The latter arrangements have long been known to have duals that are partial cubes [6]. The weak dual defined below will also turn out to be a partial cube.

Definition 3.6 (Domain $D(G)$ of G, (bounded) face of $\mathcal{E}(G)$, adjacent faces)
The domain $D(G)$ of G is the set of points covered by the vertices, edges and faces of G. A face of $\mathcal{E}(G)$ is an (open and bounded) connected component (in $\mathbb{R}^2$) of $D(G) \setminus \overline{\mathcal{E}(G)}$, where $\overline{\mathcal{E}(G)}$ denotes the set of points covered by the paths in $\mathcal{E}(G)$. Two faces $f \neq f'$ of $\mathcal{E}(G)$ are adjacent if their boundaries share more than one point.

Definition 3.7 (Weak dual Q of $\mathcal{E}(G_w)$)
Let G_w be well-arranged. Then a plane graph $Q = Q(\mathcal{E}(G_w))$ is a weak dual of $\mathcal{E}(G_w)$ if each (bounded) face of $\mathcal{E}(G_w)$ contains exactly one vertex of Q and if two vertices of Q are connected by an edge of Q if and only if the faces around the vertices are adjacent in the sense of Definition 3.6.

Due to the intersection pattern of the embedded alternating paths in G_w's faces, as specified in Section 3.2 and illustrated in Figure 2, there are the following three kinds of vertices in $V(Q)$:

Definition 3.8 (Primal, intermediate and star vertex of Q)

- ***Primal vertices:*** *Vertices which represent a face that contains a (unique) vertex v of G_w in its interior or on its boundary. If v sits on the boundary of the face, it does not sit on an alternating path but only on the boundary of $D(G_w)$ (for an example see the vertex on the upper left in Figure 1b). Thus we may let v represent the face and, more generally, interpret the vertex set of G_w as a subset of the vertex set of Q.*
- ***Intermediate vertices****: The neighbors of the vertices of the first kind.*
- ***Star vertices*** *The remaining vertices.*

For an example of a weak dual see Figure 3, where the black, gray and white vertices correspond to the primal, intermediate, and star vertices. Also note that Q is tripartite as no two vertices of the same color are connected by an edge.

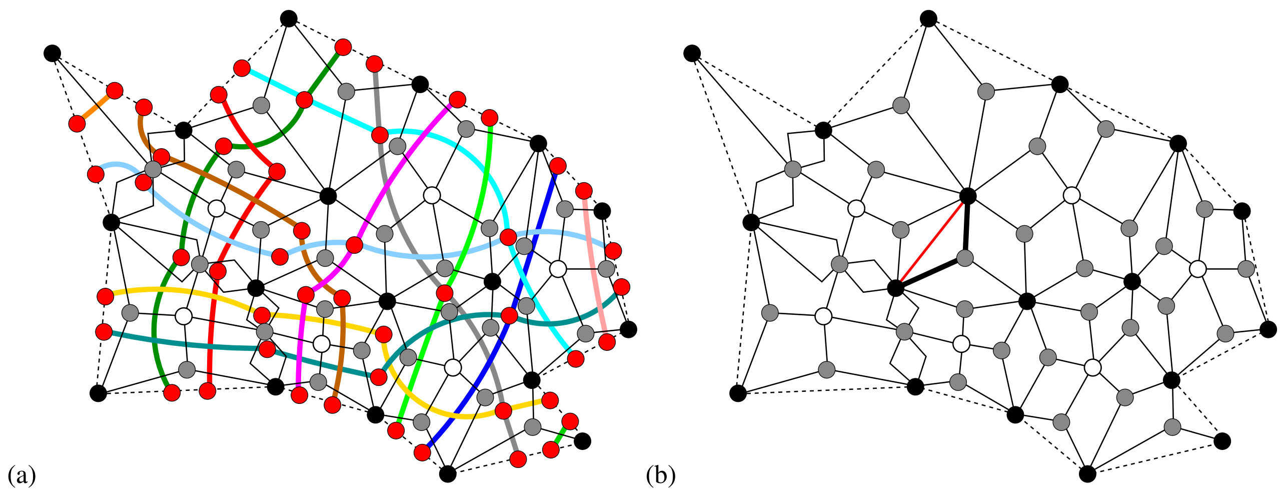


Figure 3: (a) Arrangement of alternating paths: Red vertices and thick colored solid lines. Weak dual Q: Black, gray and white vertices, thin black solid lines. The black, gray and white vertices are the primal, intermediate and star vertices, respectively. The dashed polygonal line delimits the domain of the primal graph. (b) Weak dual Q only. The red edge, however, is an edge of G. The path formed by the two bold black edges is an example of a path in Q of length two that connects two primal vertices that are adjacent in G via an intermediate vertex in Q.

4 Well-arranged plane graphs are partial half-cubes

Let G_w be a well-arranged plane graph, *i. e.*, $\mathcal{E}(G_w)$ is an arrangement of alternating paths. Also, let Q be a weak dual of $\mathcal{E}(G_w)$. In this section we prove that Q is a partial cube and that G_w is a partial half-cube. For a good overview on partial cubes see [11].

Definition 4.1 (Isometric subgraph, partial cube, (partial) half-cube)
A subgraph $S = (V_S, E_S)$ of a (not necessarily plane) graph H is an isometric subgraph *of H if $d_S(u, v) = d_H(u, v)$ for all $u, v \in V_S$. A* partial cube *is an isometric subgraph of a hypercube. Moreover, a* half-cube *is a bipartite half of a hypercube and a* partial half-cube *an isometric subgraph of a half-cube.*

Due to the structure of the hypercube, Q is a partial cube if and only if we can assign labels that are binary vectors of equal length (indicating a hypercube's corners) to the vertices of Q such that the distance between any pair of vertices in Q is equal to the Hamming distance of the corresponding binary vectors. We denote the Hamming distance of two binary vectors b and b' by $h(b, b')$ and the label of a vertex v of Q by $l(v)$.

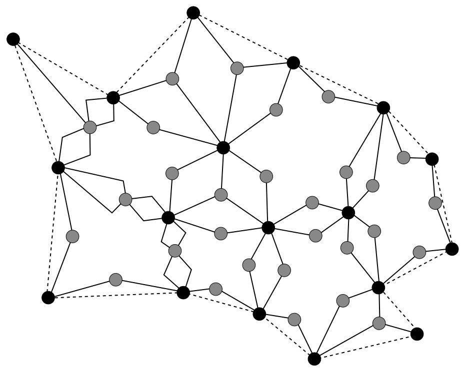


Figure 4: Isometric subgraph Q' of the weak dual Q shown in Figure 3b. The subgraph is obtained by deleting all white vertices of Q.

The length n of any binary vector $l(v)$ equals the number of paths in $\mathcal{E}(G_w)$, and the entries of $l(v)$ indicate v's position with respect to the paths in $\mathcal{E}(G_w)$. Specifically, we start by numbering the paths in $\mathcal{E}(G_w)$ from one to n, which yields the paths $P_1, \ldots, P_n$. For each $1 \leq i \leq n$ we then select one component of $D(G_w) \setminus P_i$. Finally, we set the ith entry of $l(v)$ to one if the face represented by v is in the selected component of $D(G_w) \setminus P_i$ (zero otherwise).

Theorem 4.2 *Q, the weak dual of $\mathcal{E}(G_w)$, is a partial cube.*

In order to prove that a well-arranged plane graph is an isometric subgraph of a half-cube, we focus on a subgraph Q' of a weak dual Q of $\mathcal{E}(G_w)$ first. For an example of Q', see Figure 4.

Lemma 4.3 *Let Q' denote the plane graph obtained from Q by deleting all star vertices (cf. Definition 3.8, Figure 4). Then Q' is an isometric subgraph of Q and thus a partial cube.*

Theorem 4.4 *Any well-arranged plane graph G_w is a partial half-cube.*

Proofs of Theorem 4.2, Lemma 4.3 and Theorem 4.4 can be found in the appendix.

The fact that G_w is a partial half-cube implies that it can be partitioned naturally into convex subgraphs.

5 Partitions of plane graphs from well-arranged subgraphs

We go back to the setting before Section 3.3, where a plane graph G is not necessarily well-arranged, *i. e.,* where $\mathcal{E}(G)$ is not necessarily an arrangement of alternating paths. In Section 5.1 we introduce a method to find a well-arranged subgraph of G. The latter is used in Section 5.2 to find a collection of plane curves for G that has Property 1.1.

5.1 Well-arranged subgraphs

In order to find a well-arranged subgraph of a non-well-arranged plane graph G, we first tie the violations to well-arrangedness to certain faces of G. To this end, let $P_1, \ldots, P_n$ denote the paths in $\mathcal{E}(G)$. In order for $\mathcal{E}(G)$ to be an arrangement of alternating paths, any P_i must not cross itself and may cross any $P_j \neq P_i$ at most once.

The first condition is violated if and only if there exists a face F of G such that the self-intersection of P_i occurs in $F \cup E(F)$. Such a face F is called *problematic because of a self-intersection*. The second condition is violated if and only if there exist faces $F_a \neq F_b$ of G such that

- F_a and F_b are crossed by P_i,
- $F_a \cup E(F_a)$ and $F_b \cup E(F_b)$ are intersected by P_j, and
- $E(F_a) \cap E(F_b) \cap P_j = \emptyset$.

The last condition is important because it prevents P_i from crossing P_j only once (on $E(F_a) \cap E(F_b)$). If P_i is a cycle with F_a and F_b as above, any edge of P_i is called *problematic because of multiple intersections*. If P_i is not a cycle, let

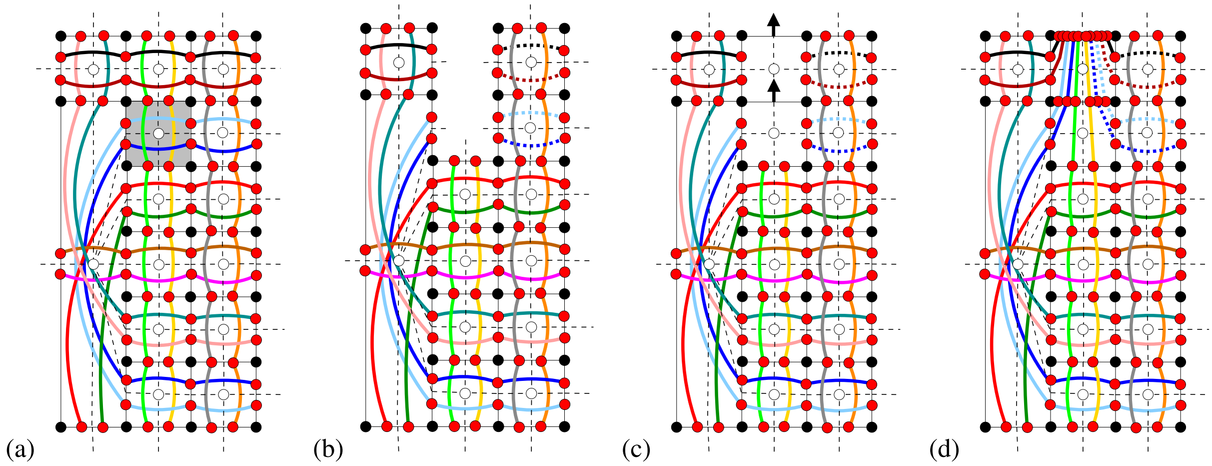


Figure 5: (a) Primal graph G: Black vertices, thin solid edges. Dual graph: White vertices, dashed edges. Maximal embedded alternating paths: Red vertices, thick solid colored lines. All faces of G that are crossed by the light blue or the dark blue path, *e. g.,* the shaded face, are problematic because of multiple intersections. (b) The shaded face in (a) is fused with the unbounded face. Thus some embedded alternating paths, *e. g.,* the dark blue path, come apart. The new subgraph of G is well-arranged. (c) Part of gradient vector field (see the two arrows in the upper part) that specifies how the shaded face is fused with the unbounded face. (d) Collection of plane curves for G that has Property 1.1. **See appendix for enlarged versions of the figures.**

$S := (F_1, \ldots, F_n)$ denote the sequence of G's faces that are crossed by P_i as P_i is traversed from one terminal vertex to the other. Then any face between F_a and F_b in S, including F_a and F_b, is called *problematic because of multiple intersections*. To summarize, G is not well-arranged if and only if it has problematic faces. For an example of a problematic face, see Figure 5a.

In the following we form subgraphs of G that have fewer and fewer problematic faces in an iterative process. The idea is to fuse problematic faces with the unbounded face F_∞ via "straight paths" of intermediate faces, if any. Specifically, we first compute $d_{G^*}(u^*, u^*_\infty)$ for any vertex u^* of G^* (recall that u^*_∞ is the vertex representing F_∞). We then form pairs (u^*, e^*), where $u^* \neq u^*_\infty$ and $e^* = \{u^*, u^*_{-1}\}$ for some u^*_{-1} such that

$$d_{G^*}(u^*_{-1}, u^*_\infty) = d_{G^*}(u^*, u^*_\infty) - 1 \quad (1)$$

These pairs specify shortest paths from any vertex of G^* to u^*_∞ and, by duality, a shortest path $(u^* = u^*_0, e^*_1, u^*_1, \ldots, e^*_k, u^*_k = u^*_\infty)$ corresponds to a sequence $(F = F_0, e_1, F_1, \ldots, e_k, F_k = F_\infty)$ of faces and edges of G. If F is a problematic face, we remove the edges $e_1, \ldots, e_k$ and thus fuse F and the intermediate faces F_i with F_∞ (see Figure 5b). If this process creates edges bounding only one face, we remove them, too. These edges are easily detected by the fact that their duals are self-loops.

Technically, we encode all pairs using a single gradient vector field (GVF) [8]. Specifically, the pairing of u^* with $e^* = \{u^*, u^*_{-1}\}$ gives rise to a vector (arrow) pointing from the dual edge of e^* into the face dual to u^*_{-1}. We refer to such a GVF as *steepest descent* GVF. The arrows in Figure 5c belong to a steepest descent GVF and specify how the shaded face in Figure 5a is fused with F_∞. For any face $F \neq F_\infty$ of G there exists exactly one edge in $E(F)$ that is the source of an arrow. Thus the fusion of the problematic faces with F_∞ can only result in a connected subgraph of G.

The entire scheme for finding a subgraph with no problematic faces, *i. e.,* a well-arranged subgraph, is shown in Algorithm 1. We can conclude:

Theorem 5.1 *Algorithm 1 computes a well-arranged subgraph of G in time $O(|E|)$.*

For a proof see the appendix.

5.2 New collections of plane curves for partitioning G

In the previous section we have determined a well-arranged subgraph G_w of G. Let F^w_∞ be the unbounded face of this subgraph G_w. An edge e of G_w that bounds F^w_∞ may or may not bound F_∞. In the latter case exactly two paths from $\mathcal{E}(G_w)$,

Algorithm 1 Find a well-arranged subgraph of a plane graph G

1. Compute a steepest descent GVF of G.
2. Define a global 1D array a with indices $1, \dots, |E|$ (note that $|E|$ is an upper bound for the number of alternating paths). Store the current number of paths in the variable n, and set all entries of array a to zero.
3. Traverse P_1 (in any direction). If P_1 is a cycle, start anywhere. Otherwise start at one of the terminal vertices. Update array a such that, at all times, $a[j]$ indicates the number of intersections with P_j so far. If $a[1] = 1$ or $a[j] \geq 2$ for some $j > 1$, a problematic face F has been found. If the intersection causing the last update of array a had occurred on an edge e of $E(F)$, all paths entering F through e intersect e at a unique point p (see Section 3.2). Split p into $p_1, p_2 \in e$ and change the embedding of the two paths intersecting at p such that they now *end* at p_1 and p_2 and such that no new intersections of the two paths are introduced.
   In any case the part of P_1 traversed so far, *i. e.,* the part up to the edge through which F was entered, becomes the new path P_{n+1}. Assign the new label $n + 1$ to each edge of the new path by going backwards on former P_1 and, in the same pass, reset array a. Finally, increment n by one.
4. Fuse F with F_∞.
5. Continue the traversal of P_1, find new problematic faces (if any), proceed as above and fuse the new problematic faces with F_∞.
6. Proceed with $P_2, \dots P_n$ as with P_1. The only difference is that some of the faces traversed by P_i may now be missing due to fusions. Before running into a missing face, assign the new label $n + 1$ to each edge of the latest section of P_i, reset array a and increment n by one. Then proceed at the next existing face crossed by P_i.
7. Determine all alternating paths in the remaining subgraph from scratch.

say P_1 and P_2, will end at e. The steepest descent GVF defined in the previous section indicates a path from e to F_∞ through bounded faces of G. We now extend P_1 and P_2 through these faces. Note that, as we follow the GVF through faces of G, more and more extensions of alternating paths may "flow" into the fused face. For an example see Figure 5d.

The GVF, however, guarantees that all extensions can be embedded without intersections. Indeed, because of the GVF, all extensions through a face F must leave F through the same edge, denoted by e_0. Let $(e_0, e_1, \dots, e_k)$ be the sequence of edges bounding F, which starts at e_0 and which is oriented, say, clockwise. We can then draw the extensions that enter F through e_1 (and leave F through e_0) as a bundle with no intersections that yields a simply connected region R in F which is still bounded by all e_i with $i > 1$. Thus we can draw the extensions entering F through e_2 as a bundle with no intersections that yields a simply connected region R' in F which is still bounded by all e_i with $i > 2$, and so on.

Extending all paths of $\mathcal{E}(G_w)$ yields a new collection $\mathcal{E}'(G)$, which has Property 1.1. Indeed, the paths in $\mathcal{E}'(G)$ do not intersect themselves because $\mathcal{E}(G_w)$ is an arrangement of alternating paths and because the extensions of the paths from $\mathcal{E}(G_w)$ are free of self-intersections, too (see above). This yields Item 1 of Property 1.1. Moreover, no pair of paths $P_1 \neq P_2 \in \mathcal{E}'(G)$ can intersect outside $D(G_w)$, and inside G_w they can intersect at most once because $\mathcal{E}(G_w)$ is an arrangement of alternating paths. This yields Item 2 of Property 1.1. Item 3 of Property 1.1 is fulfilled for any face in $D(G_w)$ because $\mathcal{E}(G_w)$ is an arrangement of alternating paths, and it is fulfilled for any face outside $D(G_w)$ because here the paths from $\mathcal{E}(G_w)$ have been extended along a steepest descent GVF. Finally, Item 4 of Property 1.1 is a consequence of the fact that any path in $\mathcal{E}'(G)$ must go through an edge of G_w.

The transformation from $\mathcal{E}(G_w)$ to $\mathcal{E}'(G)$ is guided by the steepest descent GVF which always uniquely determines the next face through which an extension must run. Hence, all extensions can be done in optimal time $O(m_{\mathcal{E}'(G)})$, where $m_{\mathcal{E}'(G)}$ is the number of edges in all paths of $\mathcal{E}'(G)$. Since the time complexity of constructing $\mathcal{E}(G)$ in the first place is optimal $\mathcal{O}(|E(G)|)$, and since the time for constructing G_w is optimal $\mathcal{O}(|E(G)|)$ (see Theorem 5.1) we have proved the following.

Theorem 5.2 *Let $m_{\mathcal{E}'(G)}$ be the number of edges in $\mathcal{E}'(G)$. A collection of maximal embedded paths that partitions G and has Property 1.1 can be computed in optimal time $\mathcal{O}(\max\{|E(G)|, m_{\mathcal{E}'(G)}\})$.*

6 Conclusions

In this paper we have combined two geometric concepts for the purpose of partitioning a plane graph G. The first concept is analogous to partitioning $\mathbb{R}^n$ by hyperplanes into (convex) halfspaces. This concept yields convex partitions of a well-arranged subgraph G_w of G. The second concept consists of using a steepest descent gradient vector field for extending every path in G_w. The collection $\mathcal{E}'(G)$ of extended paths admits Property 1.1. This makes the paths in $\mathcal{E}'(G)$ or even entire subcollections of $\mathcal{E}'(G)$ promising candidates for small cuts when solving $\mathcal{NP}$-hard constrained partitioning problems.

In future work we would like to a obtain tight bound on $m_{\mathcal{E}'(G)}$ in Theorem 5.2. Also, a further exploration of the concept of well-arranged subgraphs is of high interest, in particular concerning the size of these subgraphs.

Acknowledgments We thank P. Sanders and C. Schulz for helpful discussions on the topic.

References

[1] D. Artigas, S. Dantas, M. Dourado, and J. Szwarcfiter. Partitioning a graph into convex sets. *Discrete Mathematics*, 311(17):1968 – 1977, 2011.

[2] A. Björner, M. Las Vergnas , B. Sturmfels, N. White, and G. Ziegler. *Oriented Matroids*. Encyclopedia of Mathematics and its Applications, 46. Cambridge University Press, second edition, 1999.

[3] S. Chepoi, M. Deza, and V. Grishukhin. Clin dâĂŹoeil on L1-embeddable planar graphs. *Discrete Applied Mathematics*, 80:3–19, 1997.

[4] D. Delling, A. Goldberg, I. Razenshteyn, and R. Werneck. Exact combinatorial branch-and-bound for graph bisection. In *Proc. 14th SIAM Meeting on Algorithm Engineering and Experiments (ALENEX'12)*, pages 30–44. SIAM, 2012.

[5] D. Delling, A. V. Goldberg, I. Razenshteyn, and R. F. F. Werneck. Graph partitioning with natural cuts. In *Proc. 25th IEEE International Symposium on Parallel and Distributed Processing (IPDPS'11)*, pages 1135–1146, 2011.

[6] D. Eppstein. Cubic Partial Cubes from Simplicial Arrangements. *The electronic journal of combinatorics*, 13:#R79, 2006.

[7] C. M. Fiduccia and R. M. Mattheyses. A linear-time heuristic for improving network partitions. In *Proceedings of the 19th Conference on Design automation (DAC'82)*, pages 175–181. IEEE Press, 1982.

[8] R. Forman. A user's guide to discrete Morse theory. preprint, 2001.

[9] R. Lipton and R. Tarjan. A separator theorem for planar graphs. *SIAM Journal on Applied Mathematics*, 36(2):177–189, 1979.

[10] H. Meyerhenke, B. Monien, and T. Sauerwald. A new diffusion-based multilevel algorithm for computing graph partitions. *Journal of Parallel and Distributed Computing*, 69(9):750–761, 2009. Best Paper Awards and Panel Summary: IPDPS 2008.

[11] S. Ovchinnikov. Partial cubes: Structures, characterizations, and constructions. *Discrete Mathematics*, 308:5597âĂŞ–5621, 2008.

[12] H. Räcke. Optimal hierarchical decompositions for congestion minimization in networks. In *Proc. 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 255–264, 2008.

[13] K. Schloegel, G. Karypis, and V. Kumar. Graph partitioning for high performance scientific simulations. In *The Sourcebook of Parallel Computing*, pages 491–541. Morgan Kaufmann, 2003.

[14] J. Sherman. Breaking the multicommodity flow barrier for O($\sqrt{\log n}$)-approximations to sparsest cut. In *Proc. 50th IEEE Symp. on Foundations of Computer Science (FOCS '09)*, pages 363 –372, Oct. 2009.

A Appendix

A.1 Proof of Theorem 4.2

It suffices to show that $d_Q(u,v) = h(l(u), l(v))$ for any pair $u \neq v \in V$.

Since on any path of length k from u to v in Q it holds that $h(l(u), l(v)) \leq k$, we have $d_Q(u,v) \geq h(l(u), l(v))$.

We may assume $u \neq v$ (the case $u = v$ is trivial as both distances are 0). To see that $d_Q(u,v) = h(l(u), l(v))$, by induction it suffices to show that u has a neighbor u' such that $h(l(u'), l(v)) < h(l(u), l(v))$ (because then there also exists u'' such that $h(l(u''), l(v)) < h(l(u'), l(v))$ and so on until v is reached in $h(l(u), l(v))$ steps).

Indeed, this follows from the case distinction below. F_u stands for the face of $\mathcal{E}(G_w)$ that is represented by u, and $I(u)$ denotes the set of indices of paths in $\mathcal{E}(G_w)$ that bound F_u.

1. If u has only one neighbor u', then $I(u) = \{k\}$ for some k, and the only vertex in one of the components of $D(G_w) \backslash P_k$ is u. For an example see the black vertex in the upper left corner of Figure 1b. Since the binary labels of u and u' differ only at position k, and since the binary labels of u' and v agree at position k, it must hold that $h(l(u'), l(v)) < h(l(u), l(v))$.

2. If u has at least two neighbors, we first assume that none of the P_k with $k \in I(u)$ cross each other. Then u is uniquely determined by the entries at the positions given by $I(u)$. Indeed, F_u is then bounded by non-intersecting and non-self-intersecting paths in $\mathcal{E}(G_w)$ that go from a point on the border of $D(G_w)$ to another point on the border of $D(G_w)$). Hence only a vertex inside F_u can have the same entries as u at the positions given by $I(u)$. Thus, since u is the only vertex in F_u and since $u \neq v$, the labels of u and v must differ at a position in $I(u)$, and we are done.

3. The remaining case is that u has at least two neighbors and there exists at least one pair $(i,j) \in I(u) \times I(u)$, $i \neq j$, such that P_i crosses P_j. Let C denote the set of all such pairs. For any pair $(i,j) \in C$ the path P_i crosses the path P_j exactly once, because $\mathcal{E}(G_w)$ is an arrangement of alternating paths. Thus P_i and P_j subdivide $D(G_w)$ into four regions, each of which is characterized by one of the four 0/1 combinations of vertex label entries at i and at j. We may assume that v is contained in the same region as u for each pair $(i,j) \in C$ (otherwise we choose u' on the other side of P_i or P_j and are done). The intersection of all these regions, one region per pair in C, is denoted by R.

   If all pairs of $I(u) \times I(u)$, $i \neq j$, are in C, we are done. Indeed this means that $R = F_u$ and thus that u is uniquely determined by the entries at the positions given by $I(u)$. We can then proceed as above. The remaining case is that there exist $i \in I(u)$ such that P_i does not intersect any P_j with $j \in I(u), j \neq i$. Let the set of these indices be denoted by $I'(u)$. In particular the faces of $\mathcal{E}(G_w)$ that are contained in R are separated by the paths P_i with $i \in I'(u)$. Recall that we assumed $u \neq v \in R$, *i. e.,* u and v are contained in different faces of R. Since the paths P_i with $i \in I'(u)$ do not cross each other, the entries of u's and v's labels differ at all positions in $I'(u)$, and u' with $h(l(u'), l(v)) < h(l(u), l(v))$ can be reached from u by crossing a single path P_i with $i \in I'(u)$.

A.2 Proof of Lemma 4.3

Recall that the primal vertices of Q are precisely the vertices of G_w. Due to the intersection pattern of the embedded alternating paths, as specified in Section 3.2, any edge $e = \{u_1, u_2\}$ of G_w gives rise to at least one path of length two in Q that connects u_1 and u_2 via an intermediate grey vertex of Q (see Figure 3b).

Now let P be a path that connects vertices u and v on the boundary of a face F of G_w that is not longer than the path that connects u and v in the opposite direction around F. Then P gives rise to a path P_Q on Q that is twice as long as P (since it runs over intermediate vertices). Due to the intersection pattern of the embedded alternating paths in F, P_Q crosses any path in $\mathcal{E}(G_w)$ at most once. Since Q is a partial cube (Theorem 4.2), this means that P_Q is a shortest path from u to v in Q [11]. Consequently, in order to follow a shortest path in Q, the traversal of black primal and grey intermediate vertices is sufficient. Thus if we delete all white star vertices, we obtain Q' and preserve all shortest path distances as in Q. Also, since isometric subgraphs of partial cubes are partial cubes as well, Q' is a partial cube.

A.3 Proof of Theorem 4.4

Analogous to the argument in the proof of Lemma 4.3, the length of each shortest path in G_w has exactly half the distance of the corresponding shortest path in Q'. As before with Q, the black primal vertices of Q' form precisely $V(G_w)$. Moreover, Q' is bipartite and the black vertices form one partition. Thus, since Q' is a partial cube, G_w is a partial half-cube, *i. e.,* an isometric subgraph of a half-cube.

A.4 Proof of Theorem 5.1

Correctness follows from the fact that all problematic faces have been fused with F_∞ when the algorithm terminates, as shown in Section 5.1. The first two steps and the last one can clearly be done in $O(|E|)$ time. The claim now follows from the fact that the traversals of the paths in $\mathcal{E}(G)$ are such that any edge on any alternating path is traversed at most twice, and from the fact that the number of fusions affecting one face and one neighboring face cannot exceed the number of edges in G.

A.5 Figure 5 enlarged