Progress on Partial Edge Drawings

Till Bruckdorfer[1], Sabine Cornelsen[2], Carsten Gutwenger[3], Michael Kaufmann[1], Fabrizio Montecchiani[4], Martin Nöllenburg[5], and Alexander Wolff[6]

[1] Universität Tübingen, Germany
[2] Universität Konstanz, Germany
[3] Universität Dortmund, Germany
[4] Università degli Studi di Perugia, Italy
[5] Institut für Theoretische Informatik, KIT, Germany
[6] Lehrstuhl für Informatik I, Universität Würzburg, Germany

Abstract. Recently, a new way of avoiding crossings in straight-line drawings of non-planar graphs has been investigated. The idea of *partial edge drawings* (PED) is to drop the middle part of edges and rely on the remaining edge parts called *stubs*. We focus on a symmetric model (SPED) that requires the two stubs of an edge to be of equal length. In this way, the stub at the other endpoint of an edge assures the viewer of the edge's existence. We also consider an additional homogeneity constraint that forces the stub lengths to be a given fraction δ of the edge lengths (δ-SHPED). Given length and direction of a stub, this model helps to infer the position of the opposite stub.

We show that, for a fixed stub–edge length ratio δ, not all graphs have a δ-SHPED. Specifically, we show that K_{241} does not have a $1/4$-SHPED, while bandwidth-k graphs always have a $\Theta(1/\sqrt{k})$-SHPED. We also give bounds for complete bipartite graphs. Further, we consider the problem MAXSPED where the task is to compute the SPED of maximum total stub length that a given straight-line drawing contains. We present an efficient solution for 2-planar drawings and a 2-approximation algorithm for the dual problem.

1 Introduction

In the layout of graphs, diagrams, or maps, one of the central problems is to avoid the interference of elements such as crossing edges in graph drawings or overlapping labels on maps. This is a form of *visual clutter*. Clutter avoidance is the objective of a large body of work in graph drawing, information visualization, and cartography. In this paper, we treat a specific aspect of clutter avoidance; we focus on *completely* removing edge crossings in straight-line drawings of non-planar graphs. Clearly, this is not possible in any of the traditional graph drawing styles that insist on connecting the geometric representations of two adjacent vertices (e.g., small disks) by a closed Jordan curve (e.g., segments of straight lines). In such drawings of non-planar graphs, some pairs of edge representations must cross (or overlap). This is a serious problem when displaying dense graphs.

Previous Work. Becker et al. [2] have taken a rather radical approach to escape from this dilemma. They wanted to visualize network overload between the 110 switches of

W. Didimo and M. Patrignani (Eds.): GD 2012, LNCS 7704, pp. 67–78, 2013.

the AT&T long distance telephone network in the U.S. on October 17, 1989, when the San Francisco Bay area was hit by an earthquake. They used straight-line segments to connect pairs of switches struck by overload; the width of the segments indicated the severeness of the overload. Due to the sheer number of edges of a certain width, the underlying map of the U.S. was barely visible. They solved this problem by drawing only a certain fraction (roughly 10%) of each edge; the part(s) incident to the switch(es) experiencing the overload. We call these parts the *stubs* of an edge. The resulting picture is much clearer; it shows a distinct east–west trend among the edges with overload.

Peng et al. [15] used splines to bundle edges, e.g., in the dense graph of all U.S. airline connections. In order to reduce clutter, they increase the transparency of edges towards the middle. They compared their method to other edge bundling techniques [12,10], concluding that their method, by emphasizing the stubs, is better in revealing directional trends.

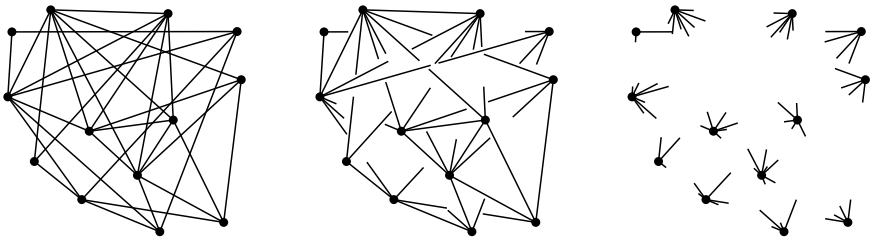
Burch et al. [6] recently investigated the usefulness of partial edge drawings of *directed* graphs. They used a single stub at the source vertex of each edge. They did a user study (with 42 subjects) which showed that, for one of the three tasks they investigated (identifying the vertex with highest out-degree), shorter stubs resulted in shorter completion times *and* smaller error rates. For the two other tasks (deciding whether a highlighted pair of vertices is connected by a path of length one/two) the error rate went up with decreasing stub length; there was just a small dip in the completion time for a stub–edge length ratio of 75%.

A similar, but less radical approach, is the use of edge *casing*. Eppstein et al. [8] have investigated how to optimize several criteria that encode the above–below behavior of edges in given graph drawings. They introduce three models (i.e., legal above–below patterns) and several objective functions such as minimizing the total number of above–below switches or the maximum number of switches per edge. For some combinations of models and objectives, they give efficient algorithms, for one they show NP-hardness; others are still open. Edge casings were re-invented by Rusu et al. [17] with reference to Gestalt principles.

Dickerson et al. [7] proposed confluent drawings to avoid edge crossings. In their approach, edges are drawn as locally monotone curves; edges may overlap but not cross.

We build on and extend the work of Bruckdorfer and Kaufmann [5] who formalized the problem of partial edge drawings (PEDs) and suggested several variants. A PED is a straight-line drawing of a graph in which each edge is divided into three segments: a middle part that is not drawn and the two segments incident to the vertices, called stubs, that remain in the drawing. In this paper, we require all PEDs to be crossing-free, i.e., that no two stubs intersect. We consider stubs relatively open sets, i.e., to not contain their endpoints. In the symmetric case (SPED), both stubs of an edge must be the same length; in the homogeneous case (HPED), the ratio of stub length over edge length is the same over all edges. A δ-SHPED is a symmetric homogeneous PED with ratio δ.

Bruckdorfer and Kaufmann showed, among others, that K_n (and thus, any n-vertex graph) has a $1/\sqrt{4n/\pi}$-SHPED. They also proved that the j-th power of any subgraph of a triangular tiling is a $1/(2j)$-SHPED. They introduced the optimization problem MAXSPED where the aim is to maximize the total stub length (or *ink*) in order to turn a given geometric graph into a SPED. They presented an integer linear program



(a) with crossings (b) as a maxSPED (c) as a maxSHPED

Fig. 1. Various drawings of a 13-vertex graph

for MAXSPED and conjectured that the problem is NP-hard. Indeed, there is a simple reduction from PLANAR3SAT [13]. Figure 1b depicts a maxSPED of the straight-line drawing in Fig. 1a, i.e., a SPED that is a solution to MAXSPED. We have slightly shrunken the stubs in the maxSPED so that they do not touch. For comparison, Fig. 1c depicts a SHPED with maximum ratio δ.

Contribution. In this paper, we focus on the symmetric case. A pair of stubs of equal length pointing towards each other at the opposite endpoints of an edge is, for the viewer of a SPED, a valuable witness that the connection actually exists. If the drawing is additionally homogeneous, finding the other endpoint of a stub is made easier since its approximate distance can be estimated from the stub length.

- We show that not all graphs admit a 1/4-SHPED; see Sect. 2. Indeed, K_n does not have a 1/4-SHPED for any $n > 240$. If we restrict vertices to be mapped to points in convex and one-sided convex position, the bound drops to 22 and 16, respectively. Our proof technique carries over to other values of the stub–edge length ratio δ.
- Recall that Bruckdorfer and Kaufmann [5] showed that K_n (and thus, any n-vertex graph) has a $1/\sqrt{4n/\pi}$-SHPED. We improve their result for specific graph classes, namely for complete bipartite graphs and for bandwidth-k graphs; see Sect. 3. The latter we show to have $\Theta(1/\sqrt{k})$-SHPEDs independently of their sizes.
- Then we turn to the optimization problem MAXSPED; see Sect. 4. For the class of 2-planar graphs, we can solve the problem efficiently; given a 2-planar drawing of a 2-planar graph with n vertices, our algorithm runs in $O(n \log n)$ time. For general graphs, we have a 2-approximation algorithm with respect to the dual problem MINSPED: minimize the amount of ink that has to be erased in order to turn a given drawing into a SPED.

Notation. In this paper, we always identify the vertices of the given graph with the points in the plane to which we map the vertices. The graphs we consider are undirected; we use uv as shorthand for the edge connecting u and v. If we refer to the stub uv then we mean the piece of the edge uv incident to u; the stub vu is incident to v.

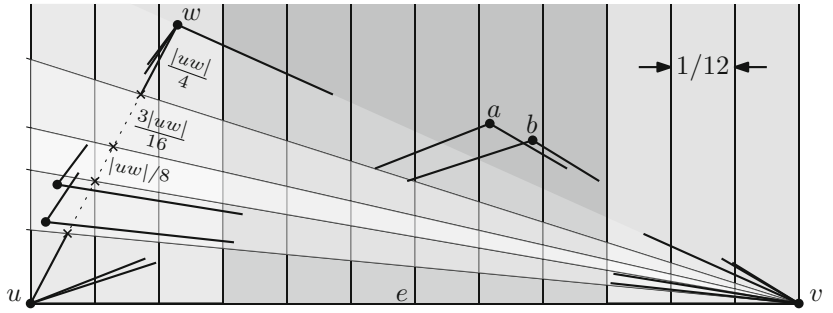


Fig. 2. Sketch of the argument why no 17 points in one-sided convex position can be used to embed K_{17} as a $1/4$-SHPED

2 Upper Bounds for Complete Graphs

In this section, we show that not any graph can be drawn as a $1/4$-SHPED. Note that $1/4$ is an interesting value since it balances the drawn and the erased parts of each edge. Yet, our proof techniques generalize to δ-SHPEDs for arbitrary but fixed $0 < \delta < 1/2$. We start with a simple proof for the scenario where we insist that vertices are mapped to specific point sets, namely point sets in convex or one-sided convex position. We say that a convex point set is *one-sided* if its convex hull contains an edge of a rectangle enclosing the point set.

Theorem 1. *There is no set of 17 points in one-sided convex position on which the graph K_{17} can be embedded as $1/4$-SHPEDs, respectively.*

Proof. We assume, to the contrary of the above statement, that there is a set P of 17 points in one-sided convex position that admits an embedding of K_{17} as a $1/4$-SHPED. Consider the edge $e = uv$ that witnesses the one-sidedness of P. We can choose our coordinate system such that $u = (0, 0)$, $v = (1, 0)$ and all other points lie above e. We split the area above e into twelve interior-disjoint vertical strips of equal width, see Fig. 2.

We first show that the union of the six innermost strips contains at most six points of P. Otherwise there would be a strip S that contains two points a and b of P. Let a be the one closer to u. Since S is one of the six innermost strips, the stub av intersects the right boundary of S (below the stub bv), and the stub bu intersects the left boundary of S (below the stub au). Point a lies above stub bu and point b lies above stub av. Hence, stubs av and bu intersect.

So at least eleven points of P must lie in the union of the three leftmost and the three rightmost strips. We may assume that the union S_{left} of the three leftmost strips contains at least six points. Let w be the rightmost point in $P \cap S_{\text{left}}$. We subdivide the edge uw into five pieces whose lengths are $1/4$, $3/16$, $1/8$, $3/16$, and $1/4$ of the length of uw. Each piece contains its endpoint that is closer to one of the endpoints of uw. The innermost piece contains both of its endpoints. Now consider the cones with apex v spanned by the five pieces of uw. We claim that no cone contains more than one point.

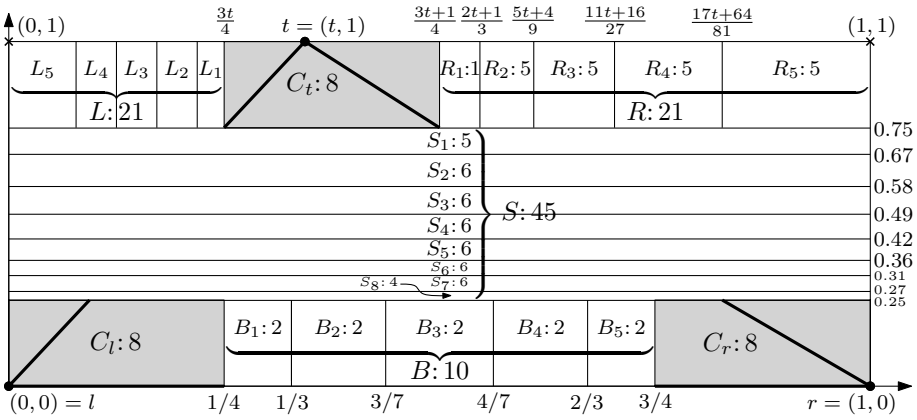


Fig. 3. Partition of the enclosing rectangle A_t into cells. We have labeled each cell or group of cells with the maximum number of points that it can contain.

Our main tool is the following. Let t be a point in $(P \cap S_{\text{left}}) \setminus \{u, w\}$. Then the stub tv intersects the right boundary of S_{left} and, hence, also the edge uw that separates $P \cap S_{\text{left}}$ from $P \setminus S_{\text{left}}$. It remains to note that in each cone, any point has a stub to u or w (whichever is further away from the cone) that intersects the boundary of the cone. □

Theorem 1 can be used to derive a first upper bound on general point sets as follows.

Corollary 1. *For any $n > \binom{30}{15}$, the graph K_n does not admit a $1/4$-SHPED.*

Proof. By a result of Erdős and Szekeres [9], any set of more than $\binom{2k-4}{k-2}$ points in general position contains a subset of k points that form a one-sided convex set. Combining this with Theorem 1 and plugging in $k = 17$ yields the claimed bound. □

We now vastly improve upon the bound of Corollary 1. Let P be the point set in the plane, and let l and r be the two points on the convex hull that define the diameter of P, which is the largest distance between any two points. We rotate P such that the line lr is horizontal and l is on the left-hand side. Now let A be the smallest enclosing axis-aligned rectangle that contains P, and let t and b be the top- and bottommost points in A, respectively. Accordingly, let A_t be the part of A above (and including) lr and let $A_b = A \setminus A_t$. We consider the two rectangles separately and assume that the interior of A_t is not empty. (In our proof we argue, for any interior point, using only its stubs towards the three boundary points l, r, and t.)

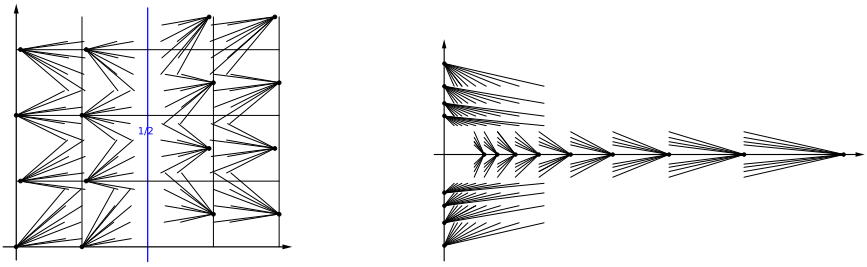
We subdivide A_t into 26 cells such that, for each point in a cell, the three stubs to l, r, and t intersect the boundary of that cell; see Fig. 3. For each cell, we prove, in the full version of this paper [4], an upper bound on the maximum number of points it can contain. Summing up these numbers (see again Fig. 3), we get a bound of 103 points in total. Since we may have a symmetric subproblem below lr, we double this number, subtract 2 because of double-counting l and r, and finally get the following theorem.

Theorem 2. *For any $n > 240$, the graph K_n does not admit a $1/4$-SHPED.*

3 Improved Bounds for Specific Graph Classes

In this section, we improve, for specific graph classes, the result of Bruckdorfer and Kaufmann [5] which says that K_n (and thus, any n-vertex graph) has a $1/\sqrt{4n/\pi}$-SHPED. In other words, K_n has a δ-SHPED if $n \leq \pi/(4\delta^2)$. We give two constructions for complete bipartite graphs and one for graphs of bounded bandwidth.

Complete Bipartite Graphs. Our first construction is especially suitable if both sides of the bipartition have about the same size. The drawing is illustrated in Fig. 4a. Note that the figure scales x- and y-axis differently.



(a) $1/4$-SHPED of $K_{8,8}$ (b) $1/4$-SHPED of $K_{8,9}$

Fig. 4. Two methods for drawing complete bipartite graphs as SHPEDs

Theorem 3. *The complete bipartite graph $K_{n,n}$ has a δ-SHPED if*

$$n \leq \left\lfloor \frac{1}{\delta} \right\rfloor \cdot \left\lfloor \left\lfloor \frac{\log 1/2}{\log(1-\delta)} \right\rfloor \right\rfloor,$$

where $\lfloor \lfloor r \rfloor \rfloor$ denotes the largest integer that is strictly less than r.

Proof. Let $k = \left\lfloor \frac{1}{\delta} \right\rfloor$ and $\ell = \left\lfloor \left\lfloor \frac{\log 1/2}{\log(1-\delta)} \right\rfloor \right\rfloor$. The latter implies that $(1-\delta)^\ell > \frac{1}{2}$.

Divide the plane at the vertical line $x = 1/2$ into two half planes, one for each side of the bipartition, to which we will refer as the right-hand side and the left-hand side. In each half plane draw the n vertices on a (perturbed) $k \times \ell$ grid. More precisely, for a horizontal line, let $\epsilon \geq 0$ such that $(1-\delta)^\ell > \frac{1}{2} + \epsilon$. Draw the vertices with x-coordinates

$$(1-\delta)^i - \epsilon \text{ and } 1 - (1-\delta)^i + \epsilon, i = 0, \ldots, \ell - 1.$$

Draw the vertices on the left-hand side with y-coordinates $0, \ldots, k-1$ and the vertices on the right-hand side with y-coordinates $0 + \sigma, \ldots, k - 1 + \sigma$ where $0 < \sigma < 1$ is chosen such that no two vertices on the right-hand side are collinear with a vertex on the left-hand side and vice versa. All edges are between a vertex on the left-hand side and a vertex on the right-hand side.

Then for any two vertices the bounding boxes of their incident stubs are disjoint up to their boundaries. Intersections of the stubs on the boundaries can be avoided by a suitable choice of ϵ. For details, see the full version of this paper [4]. □

Our second construction is especially suitable if one side of the bipartition is much larger than the other. The drawing is illustrated in Fig. 4b.

Theorem 4. *For any integers $n > 0$ and $k < \log\delta/\log(1-\delta)$, the complete bipartite graph $K_{2k,n}$ has a δ-SHPED.*

Proof. Draw the n vertices on the x-axis with x-coordinate $x_i = 1/(1-\delta)^{i-1}, i = 1,\dots,n$ and the $2k$ vertices on the y-axis with y-coordinate $y_i = 1/(1-\delta)^{i-1}, i = 1,\dots,k$ and $-y_i, i = 1,\dots,k$. All edges are between a vertex on the y-axis and a vertex on the x-axis. To show that no stubs intersect, we establish, in the full version of this paper [4], the following two properties on the regions that contain the stubs.

1. The stubs incident to $(0,\pm y_i), i = 2,\dots,k$ are in the horizontal strip bounded by $y = \pm y_i$ and $y = \pm y_{i-1}$.
2. The stubs incident to $(0,x_i), i = 2,\dots,n$ are in the rectangle bounded by $y = \pm(1-\delta)$, $x = x_{i-1}$, and $x = x_i$ (where $x_0 = 1-\delta$).

Since the stubs incident to $(0,\pm y_1)$ lie in the horizontal strip bounded by $y = \pm 1$ and $y = \pm(1-\delta)$, it follows that any two stubs are disjoint. □

Graphs of Bounded Bandwidth. Recall that the *k-circulant* graph C_n^k with n vertices and $0 \le k < n$ is the undirected simple graph whose vertex set is $\{v_0,\dots,v_{n-1}\}$ and whose edge set is $\{v_iv_j\colon |j-i| \le k\}$. When we specify the index of a vertex, we implicitly assume calculation modulo n. Note that $C_n^1 = C_n$ and $C_n^{n/2} = K_n$. We provide δ-SHPED constructions for k-circulant and bandwidth-k graphs where $\delta = \Theta\big(1/\sqrt{k}\big)$. For ease of presentation, we assume that $\sqrt{k}$ and $n/\sqrt{k}$ are integers.

First, let G be a graph of bandwidth k, i.e., the vertices of G can be ordered $v_1,\dots,v_n$ and for each edge v_iv_j it holds that $|j-i| \le k$. We draw G as a δ-SHPED as follows. We map the vertices of G to the vertices of an integer grid of $\big(n/\sqrt{k} \times \sqrt{k}\big)$ points such that the sequence of vertices $v_1,\dots,v_n$ traverses the grid column by column in a snake-like fashion, see Fig. 5a.

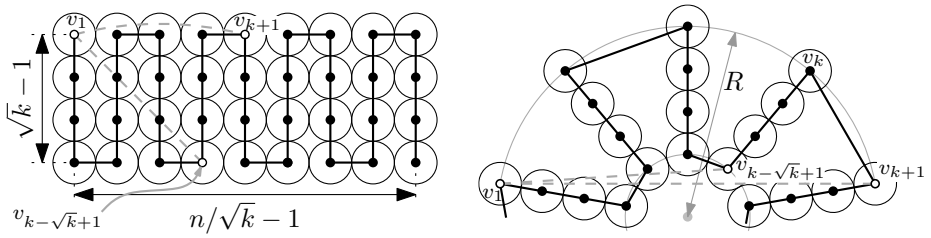
The distance from any vertex to its k-th successor is at most $\sqrt{\big(\sqrt{k}-1\big)^2 + k} < \sqrt{2k}$, see the two dashed line segments in Fig. 5a. Setting $\delta = 1/\big(2\sqrt{2k}\big)$ ensures that each stub is contained in the radius-$1/2$ disks centered at the vertex to which it is incident; see Fig. 5a. Since the disks are pairwise disjoint, so are the stubs.

For the k-circulant graph C_n^k, we modify this approach such that start and end of the snake coincide. In other words, we deform our rectangular section of the integer grid into an annulus; see Fig. 5b. We additionally assume that $n/\sqrt{k}$ is even.

The inner circle circumscribes a regular $\big(n/\sqrt{k}\big)$-gon Π of edge length 1.

We place the vertices of C_n^k on rays that go from the center of the annulus through the vertices of Π. On each ray, we place $\sqrt{k}$ vertices at distance 1 from one another, starting from the inner circle and ending at the outer circle. The sequence again traverses the stacks of vertices in a snake-like fashion.

A vertex v can be reached from its j-th ($j < k$) successor s, by traversing at most $3\sqrt{k} - 2$ segments of length 1: at most $\sqrt{k} - 1$ segments from s to the inner circle, at most $\sqrt{k}$ segments on the inner circle, and at most $\sqrt{k} - 1$ segments from the inner



(a) path case (b) circle case

Fig. 5. SHPEDs for bandwidth-k and k-circulant graphs

circle to v. Hence, the maximum distance of two adjacent vertices is less than $3\sqrt{k}$, and we can choose $\delta = 1/(6\sqrt{k})$.

Theorem 5. *Let $2 \leq k \leq n$ and assume that $\sqrt{k}$ and $n/\sqrt{k}$ are integers. Then any graph of bandwidth k has a $1/\big(2\sqrt{2k}\big)$-SHPED. If additionally $n/\sqrt{k}$ is even, the k-circulant graph C_n^k has a $1/\big(6\sqrt{k}\big)$-SHPED.*

4 Geometrically Embedded SPEDs

Bruckdorfer and Kaufmann [5] gave an integer-linear program for MAXSPED and conjectured that the problem is NP-hard. Indeed, there is a simple reduction from PLANAR3SAT [13]. In this section, we first show that the problem can be solved efficiently for the special case of graphs that admit a 2-planar geometric embedding. Then we turn to the dual problem MINSPED of minimizing the ink that has to be erased in order to turn a given drawing into a SPED.

4.1 Maximizing Ink in Drawings of 2-Planar Graphs

In this section we prove that, given a 2-planar geometric embedding Γ of a 2-planar graph G with n vertices, we can compute a maxSPED, i.e., a SPED that maximizes the total stub length, in $O(n \log n)$ time. Recall that a graph G is 2-planar if it admits a simple drawing on the plane where each edge is crossed at most twice.

Given G and Γ, we define a simple undirected graph C as follows. C has a vertex v_e for each edge e of G. Two vertices v_e and $v_{e'}$ of C are connected by an edge if and only if e and e' form a crossing in Γ. Such a graph is in general non-connected. Furthermore, since the maximum degree of C is 2, a connected component of C is either a path (possibly formed by only one edge) or a cycle.

Let C_i be a connected component of C. We define a total ordering of the vertices of C_i. Namely, if C_i is a path such an ordering is directly defined by the order of its vertices along the path (rooted at an arbitrary end vertex). If C_i is a cycle, we simply delete an arbitrary edge of the cycle, obtaining again a path and the related order. That means, if we consider the subdrawing Γ_i of Γ induced by the vertices of C_i (edges of G_i), such

a drawing is formed by an ordered sequence of edges (according to the ordering of the vertices of C_i), $e_1, \ldots, e_{n_i}$, such that e_j crosses $e_{(j+1) \bmod n_i}$ for $j = 1, \ldots, n_i - 1$ in case of a path, and $j = 1, \ldots, n_i$ in case of a cycle.

We will use the following notation: l_j is the total length of the edge e_j; x'_j is the length of the shortest stub of e_j defined by the crossing between e_{j-1} and e_j, called the backward stub; x''_j is the length of the shortest stub of e_j defined by the crossing between e_j and e_{j+1}, called the forward stub. See also Fig. 6.

Consider now the subdrawing Γ_i, and assume that $e_1, \ldots, e_{n_i}$ form a path in C_i. If $n_i = 2$, the maximum total length of the stubs is $k_{\text{opt}} = \max\{l_1 + 2x'_2, l_2 + 2x''_1\}$.

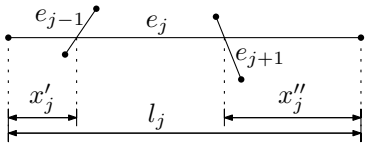


Fig. 6. Notation used in the DP

In the general case, we can process the path edge by edge, having at most three choices for each edge: (i) we can draw it entirely, (ii) we can draw only its backward stubs, or (iii) we can draw only its forward stubs. The number of choices we have at any step is influenced only by the previous step, while the best choice is determined only by the rest of the path. Following this approach, let γ_i be a maxSPED for Γ_i and consider the choice done for the first edge e_1 of the path. The total length of the stubs in γ_i, minus the length of the stubs assigned to e_1, represents an optimal solution for $\Gamma_i \setminus e_1$, under the initial condition defined by the first step, otherwise, γ_i could be improved, a contradiction. I.e., the optimality principle holds for our problem. Thus, we can exploit the following dynamic programming (DP) formulation, where $O_{\text{in}}(e_j)$ describes the maximum total length of the stubs of $e_j, \ldots, e_{n_i}$ under the choice (i) for e_j, $O'_{\text{out}}(e_j)$ describes the choice (ii) and $O''_{\text{out}}(e_j)$ describes the choice (iii).

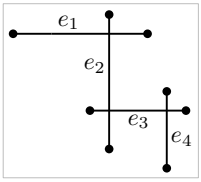
$$O_{\text{in}}(e_j) = \begin{cases} l_j + \max\{O'_{\text{out}}(e_{j+1}), O''_{\text{out}}(e_{j+1})\} & \text{if } x'_{j+1} \geq x''_{j+1}, \\ l_j + O'_{\text{out}}(e_{j+1}) & \text{if } x'_{j+1} < x''_{j+1}. \end{cases} \tag{1a}$$

$$O'_{\text{out}}(e_j) = \begin{cases} 2x'_j + \max\{O'_{\text{out}}(e_{j+1}), O''_{\text{out}}(e_{j+1})\} & \text{if } x'_j > x''_j \text{ and } x'_{j+1} \geq x''_{j+1}, \\ 2x'_j + O'_{\text{out}}(e_{j+1}) & \text{if } x'_j > x''_j \text{ and } x'_{j+1} < x''_{j+1}, \\ 2x'_j + \max\{O_{\text{in}}(e_{j+1}), O'_{\text{out}}(e_{j+1}), O''_{\text{out}}(e_{j+1})\} & \text{if } x'_j \leq x''_j. \end{cases} \tag{1b}$$

$$O''_{\text{out}}(e_j) = 2x''_j + \max\{O_{\text{in}}(e_{j+1}), O'_{\text{out}}(e_{j+1}), O''_{\text{out}}(e_{j+1})\} \tag{1c}$$

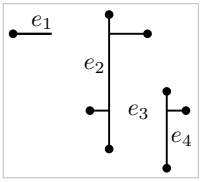
In case of a path, we store in a table the values of $O_{\text{in}}(e_j)$, $O'_{\text{out}}(e_j)$ and $O''_{\text{out}}(e_j)$, for $j = 1, \ldots, n_i$, through a bottom-up visit of the path (from e_{n_i} to e_1). Since e_1 and e_{n_i} do not cross, we have $x'_1 = l_1/2$ and $x''_{n_i} = l_{n_i}/2$. Then, the maximal value of ink is given by $k_{\text{opt}} = \max\{O_{\text{in}}(e_1), O'_{\text{out}}(e_1), O''_{\text{out}}(e_1)\}$. See Fig. 7 for an example.

In case of a cycle, we have that e_1 and e_{n_i} cross each other, thus, in order to compute the table of values we must assume an initial condition for e_{n_i}. Namely, we perform the bottom-up visit from e_{n_i} to e_1 three times. The first time we consider as initial condition that e_{n_i} is entirely drawn (choice $O_{\text{in}}(e_{n_i})$), the second time we consider only the backward stubs drawn (choice $O'_{\text{out}}(e_{n_i})$), and the third time we consider only the forward stubs drawn (choice $O''_{\text{out}}(e_{n_i})$). Every initial condition will lead to a table where, in general, we do not have all the three possible choices for e_1 (i.e., some choices are



	l	x'	x''
e_1	7	3.5	2
e_2	7	1	2
e_3	5	1	1
e_4	4	1	2

(a)



	O_{in}	O'_{out}	O''_{out}
e_4	4	2	4
e_3	7	6	6
e_2	13	9	11
e_1	16	16	17

(b)

Fig. 7. (a) A 2-planar drawing Γ and (b) a maxSPED of Γ computed by the DP algorithm

forbidden due to the initial condition). Performing the algorithm for every possible initial condition and choosing the best value yields the optimal solution k_{opt}. The algorithm described above leads to the following result.

Theorem 6. *Let G be a graph with n vertices, and let Γ be a 2-planar geometric embedding of G. A maxSPED of Γ can be computed in $O(n \log n)$ time.*

Proof. Consider the above described algorithm, based on the DP formulation defined by the set of equations (1). We already showed how this algorithm computes a maxSPED of Γ. The construction of the graph C requires time $O(m \log m)$ with a standard sweep-line algorithm for computing the $O(m)$ line-segment intersections [3]. Once C has been constructed, ordering its vertices requires $O(n_C)$ time, where $n_C \in O(m)$ is the number of vertices of C. Performing a bottom-up visit and up to three top-down visits of every path or cycle takes $O(m)$ time. Thus, the overall time complexity is $O(n \log n)$, since for 2-planar graphs $m \in O(n)$ [14]. □

We finally observe that the restricted $0/1$-MAXSPED problem for 2-planar drawings, where each edge is either drawn or erased completely, may be solved through a different approach. Indeed, we can exploit a maximum-weight SAT formulation in the CNF+($\leq$2) model, where each variable can appear at most twice and only with positive values [16]. Roughly speaking, we map each edge to a variable, with the weight of the variable equal to the length of its edge, and define a clause for each crossing. Applying an algorithm of Porschen and Speckenmeyer [16] for CNF+($\leq$2) solves $0/1$-MAXSPED in $O(n^3)$ time. However, our algorithm solves a more general problem in less time.

4.2 Erasing Ink in Arbitrary Graph Drawings

In this section, we consider the problem MINSPED, which is dual to MAXSPED. In MINSPED, we are given a graph with a straight-line drawing (i.e., a geometric graph), and the task is to erase as little of the edges as possible in order to make it a SPED.

We will exploit a connection between the NP-hard minimum-weight 2-SAT problem (MINW2SAT) and MINSPED. Recall that MINW2SAT, given a 2-SAT formula, asks for a satisfying variable assignment that minimizes the total weight of the true variables. There is a 2-approximation algorithm for MINW2SAT that runs in $O(vc)$ time and uses $O(c)$ space, where v is the number of variables and c is the number of clauses of the given 2-SAT formula [1].

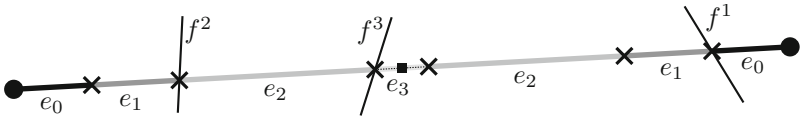


Fig. 8. Edge e is split into four pairs of edge segments; pairs are labeled equally

Theorem 7. MINSPED *can be 2-approximated in time quadratic in the number of crossings of the given geometric graph.*

Proof. Given an instance G of MINSPED, we construct an instance φ of MINW2SAT as follows. Let e be an edge of G with k crossings. Then e is split into $k+1$ pairs of edge segments $e_0, \ldots, e_k$ as shown in Fig. 8. If we order the edges that cross e in increasing order of the distance of their crossing point to the closer endpoint of e, we can assign each segment pair e_i for $i \geq 1$ to the ith edge f^i crossing e, in this order. We also say that edge f^i *induces* segment pair e_i. Any valid maximal (non-extensible) partial edge drawing of e is the union $\bigcup_{i=0}^{j} e_i$ of all pairs of edge segments up to some index $j \leq k$.

We model all pairs of (induced) edge segments as truth variables $\hat{e}_1, \ldots, \hat{e}_k$ with the interpretation that the pair e_i is *not* drawn if $\hat{e}_i = \mathit{true}$. The pair e_0 is always drawn. For $i = 1, \ldots, k$, we introduce the clause $(\neg\hat{e}_{i+1} \Rightarrow \neg\hat{e}_i) \equiv (\hat{e}_{i+1} \vee \neg\hat{e}_i)$. This models that e_{i+1} can only be drawn if e_i is drawn. Moreover, for every crossing between two edges e and f, we introduce the clause $(e_i \vee f_j)$, where e_i is the segment pair of e induced by f and f_j is the segment pair of f induced by e. This simply means that at least one of the two induced segment pairs is not drawn and thus the crossing is avoided.

Now we assign a weight $w_{e,i}$ to each variable $\hat{e}_i$, which is either the absolute length $|e_i|$ of e_i if we are interested in ink, or the relative length $|e_i|/(2|e|)$ if we are interested in relative stub lengths (δ). Then minimizing the value $\sum_{\hat{e}_i \in \mathrm{Var}(\varphi)} w_{e,i}\hat{e}_i$ over all valid variable assignments minimizes the weight of the erased parts of the edges in the given geometric graph

The 2-approximation algorithm for MINW2SAT yields a 2-approximation for the problem to erase the minimum ink from the given straight-line drawing of G. It runs in $O(vc) = O(I^2) \subseteq O(m^4)$ time since our 2-SAT formula has $O(I) \subseteq O(m^2)$ variables and clauses, where m is the number of edges of G and I is the number of intersections in the drawing of G. □

If we encode the primal problem (maximize ink) using 2SAT, we cannot hope for a similar positive result. The reason is that the tool that we would need, namely an algorithm for the problem MAXW2SAT dual to MINW2SAT would also solve maximum independent set (MIS). For MIS, however, no $(n^{1-\varepsilon})$-approximation exists unless $\mathcal{NP} = \mathcal{ZPP}$ [11].

To see that MAXW2SAT can be used to encode MIS, use a variable $\hat{v}$ for each vertex v of the given (graph) instance G of MIS and, for each edge $\{u, v\}$ of G, the clause $(\hat{u} \vee \hat{v})$. Let φ be the conjunction of all such clauses. Then finding a satisfying truth assignment for φ that maximizes the number of false variables (i.e., all variable weights are 1) is equivalent to finding a maximum independent set in G. Note that this does *not* mean that maximizing ink is as hard to approximate as MIS.

Acknowledgments. We thank Ferran Hurtado and Yoshio Okamoto for invaluable pointers to results in discrete geometry. We thank Emilio Di Giacomo, Antonios Symvonis, Henk Meijer, Ulrik Brandes, and Gašper Fijavž for helpful hints and intense discussions. We thank Jarek Byrka for the link between ink maximization and MIS. Thanks to Thomas van Dijk for drawing Fig. 1 (and implementing the ILP behind it).

References

1. Bar-Yehuda, R., Rawitz, D.: Efficient algorithms for integer programs with two variables per constraint. Algorithmica 29, 595–609 (2001)
2. Becker, R.A., Eick, S.G., Wilks, A.R.: Visualizing network data. IEEE Trans. Visual. Comput. Graphics 1(1), 16–28 (1995)
3. Bentley, J.L., Ottmann, T.A.: Algorithms for reporting and counting geometric intersections. IEEE Trans. Comput. 28(9), 643–647 (1979)
4. Bruckdorfer, T., Cornelsen, S., Gutwenger, C., Kaufmann, M., Montecchiani, F., Nöllenburg, M., Wolff, A.: Progress on partial edge drawings. Arxiv, arxiv.org/abs/1209.0830 (2012)
5. Bruckdorfer, T., Kaufmann, M.: Mad at Edge Crossings? Break the Edges! In: Kranakis, E., Krizanc, D., Luccio, F. (eds.) FUN 2012. LNCS, vol. 7288, pp. 40–50. Springer, Heidelberg (2012)
6. Burch, M., Vehlow, C., Konevtsova, N., Weiskopf, D.: Evaluating Partially Drawn Links for Directed Graph Edges. In: van Kreveld, M., Speckmann, B. (eds.) GD 2011. LNCS, vol. 7034, pp. 226–237. Springer, Heidelberg (2012)
7. Dickerson, M., Eppstein, D., Goodrich, M.T., Meng, J.Y.: Confluent drawings: Visualizing non-planar diagrams in a planar way. J. Graph Algorithms Appl. 9(1), 31–52 (2005)
8. Eppstein, D., van Kreveld, M., Mumford, E., Speckmann, B.: Edges and switches, tunnels and bridges. Comput. Geom. Theory Appl. 42(8), 790–802 (2009)
9. Erdős, P., Szekeres, G.: A combinatorial problem in geometry. Compositio Mathematica 2, 463–470 (1935)
10. Gansner, E., Hu, Y., North, S., Scheidegger, C.: Multilevel agglomerative edge bundling for visualizing large graphs. In: 4th IEEE Pacific Visual. Symp. (PacificVis 2011), pp. 187–194. IEEE Press, New York (2011)
11. Håstad, J.: Clique is hard to approximate within $n^{1-\varepsilon}$. Acta Math. 182, 105–142 (1999)
12. Holten, D., van Wijk, J.J.: Force-directed edge bundling for graph visualization. Comput. Graphics Forum 28(3), 983–990 (2009)
13. Kindermann, P., Spoerhase, J.: Private Communication (March 2012)
14. Pach, J., Tóth, G.: Graphs drawn with few crossings per edge. Combin. 17, 427–439 (1997)
15. Peng, D., Lu, N., Chen, W., Peng, Q.: SideKnot: Revealing relation patterns for graph visualization. In: 5th IEEE Pacific Visual. Symp. (PacificVis 2012), pp. 65–72. IEEE Press, New York (2012)
16. Porschen, S., Speckenmeyer, E.: Algorithms for Variable-Weighted 2-SAT and Dual Problems. In: Marques-Silva, J., Sakallah, K.A. (eds.) SAT 2007. LNCS, vol. 4501, pp. 173–186. Springer, Heidelberg (2007)
17. Rusu, A., Fabian, A.J., Jianu, R., Rusu, A.: Using the gestalt principle of closure to alleviate the edge crossing problem in graph drawings. In: 15th Int. Conf. Inform. Visual. (IV 2011), pp. 488–493. IEEE Press, New York (2011)