Minimizing the Stabbing Number of Matchings, Trees, and Triangulations[*]

Sándor P. Fekete[†]   Marco E. Lübbecke[‡]   Henk Meijer[§]

Abstract

The (axis-parallel) stabbing number of a given set of line segments is the maximum number of segments that can be intersected by any one (axis-parallel) line. This paper deals with finding perfect matchings, spanning trees, or triangulations of minimum stabbing number for a given set of vertices. The complexity of finding a spanning tree of minimum stabbing number is one of the original 30 questions on "The Open Problems Project" list of outstanding problems in computational geometry by Demaine, Mitchell, and O'Rourke.

We show $\mathcal{NP}$-hardness of stabbing problems by means of a general proof technique. For matchings, this also implies a non-trivial lower bound on the approximability. On the positive side, we propose a cut based integer programming formulation for minimizing the stabbing number of matchings and spanning trees. From the corresponding linear programming relaxation we obtain polynomial-time lower bounds and show that there always is an optimal fractional solution that contains an edge of at least constant weight. We conjecture that the resulting iterated rounding scheme constitutes a constant-factor approximation algorithm.

ACM Classification: F.2.2 Nonnumerical Algorithms and Problems.

AMS Classification: 68Q17, 68U05, 90C27.

Keywords: Stabbing number, matching, spanning tree, triangulation, complexity, linear programming relaxation, iterated rounding.

1 Introduction

Objective Functions. Many problems in combinatorial optimization, algorithmic graph theory, or computational geometry deal with minimizing the length of a desired structure: given a set of vertices, find a set of line segments of small total length, such that a certain structural condition is maintained. Among the most popular structures are spanning trees, perfect matchings, or (in a planar geometric setting) triangulations of minimum total length. Other geometric problems give rise to other objective

functions: for example, one can ask for the total turn cost between adjacent line segments; e.g., see [3].

When dealing with structural or algorithmic properties, another possible objective function is the *stabbing number*: for a given set of line segments, this is the maximum number of segments that are encountered (in their interior or at an endpoint) by any line. If we consider only axis-parallel lines, we get the *axis-parallel stabbing number*. A closely related measure defined by Matoušek [20] is the *crossing number*, which is the number of connected components of the intersection of a line with the union of line segments[1]. If there are no connected components of collinear segments (which is the case for matchings), the crossing number coincides with the stabbing number. When considering structures like triangulations, the crossing number is precisely one more than the maximum number of triangles intersected by any one line.

Stabbing problems have been considered for several years. The complexity of many algorithms in computational geometry is directly dependent on the complexity of ray shooting; as described by Agarwal [1], the latter can be improved by making use of spanning trees of low stabbing number. We will sketch some related results further down. Most previous work on stabbing and crossing problems has focused on extremal properties, and little has been known about the computational complexity of actually finding structures of low stabbing number, or possible approximation algorithms. In fact, settling the complexity of Minimum Stabbing Number for spanning trees has been one of the original 30 outstanding open problems of computational geometry on the list by Mitchell and O'Rourke [21]. (An up-to-date list is maintained online by Demaine, Mitchell, and O'Rourke [8].)

Our Contributions. We describe a general proof technique that shows $\mathcal{NP}$-hardness of minimizing the stabbing number of perfect matchings, triangulations, and spanning trees. For the case of matchings we show that it is also hard to approximate the minimum stabbing number within a factor below 6/5.

On the other hand, we present a mathematical programming framework for actually finding structures with small stabbing number. Our approach characterizes solutions to stabbing problems as integer programs (IPs) with an exponential number of cut constraints. We describe how the corresponding linear programming (LP) relaxations can be solved in polynomial time, providing empirically excellent lower bounds. Furthermore, we show that there always is an optimal fractional matching (or spanning tree) that contains an edge of weight above a lower bound of 1/3 (or 1/5 for spanning trees), allowing an iterated rounding scheme similar to the one developed by Jain for the generalized Steiner network problem [17]: compute a heuristic solution by solving a polynomial number of LPs. We conjecture that the objective function value of this heuristic solution is within a constant factor of the optimum. Our mathematical programming approach is also practically useful: as described in detail in our experimental study [12], we can optimally solve stabbing problems for instances (taken from well-known benchmark sets of other geometric optimization problems) of vertex sets up to several hundred vertices.

Our results in detail:

- We prove that deciding whether a vertex set has a perfect matching of axis-parallel stabbing number 5 is an $\mathcal{NP}$-complete problem; we also extend this result to general stabbing number.

---

[1]This should not be confused with the crossing number in graph drawing, which is the total number of crossing line segments.

- We prove that finding a triangulation of minimum axis-parallel stabbing number is an $\mathcal{NP}$-hard problem; we also extend this result to general stabbing number.

- We prove that finding a spanning tree of minimum axis-parallel stabbing number is an $\mathcal{NP}$-hard problem; we extend this result to general stabbing number, and sketch $\mathcal{NP}$-hardness proofs for minimum axis-parallel or general crossing number.

- We give an IP-based formulation for stabbing problems; the corresponding fractional LP solutions can be computed in polynomial time, providing a family of lower bounds.

- We give results on the structure of fractional vertices of the resulting LP relaxation: for matching, we show that there always is an edge with weight at least $1/5$, while for spanning trees, there always is an edge with weight greater than $1/3$. This gives way to a heuristic algorithm based on iterated rounding; we conjecture that the resulting solution values are within a constant factor of the optimum.

The vertex sets constructed in our hardness proofs make critical use of the collinearity of vertices. On the other hand, our positive (LP-based) results do *not* make any assumptions on the structure of the vertex set: they can be used for vertex sets in degenerate as well as in general position, and can be applied to any family of stabbing lines that can be evaluated by considering a subset of polynomially many representatives.

We have also performed a computational study on a diverse set of instances; the results show that our LP-based approach is good not only in theory (where we get a polynomial running time based on the ellipsoid method), but also for actually solving instances in practice (where we use the simplex method). Details are omitted from this theoretical paper; a report on the practical results can be found in [12].

Related Work. Existing work dealing with structures of low stabbing number can be divided into algorithmic applications and implications on one hand, and extremal properties on the other hand.

Agarwal [1] describes improved algorithmic solutions for problems such as ray shooting and implicit point locations queries (which by themselves have applications in polygon containment, implicit hidden surface removal, polygon placement, etc.); his main tool are spanning trees with low stabbing number. One of the theoretically best performing data structures for ray tracing in two dimensions is based on a triangulation of the polygonal scene; see Hershberger and Suri [16]: in their "pedestrian" approach to ray shooting, the complexity of a query is simply the number of triangles visited, i.e., corresponds precisely to the stabbing number. Held, Klosowski, and Mitchell [15] investigate collision detection in a virtual reality environment, again, based on "pedestrian" ray shooting. More recently, Aronov et al. [4] have performed an experimental study of the complexity of ray tracing algorithms and run-time predictors, which include average number of intersection points for a transversal line, and depth complexity.

Extremal properties of crossing numbers were considered by Welzl [26] and by Matoušek [20], who showed that any planar set of n vertices has a spanning tree with a crossing number of $O(\sqrt{n})$, and provided examples requiring a crossing number of $\Omega(\sqrt{n})$. Another variant is studied by de Berg and van Kreveld [7]: the stabbing number of a decomposition of a rectilinear polygon P into rectangles is the maximum number of rectangles intersected by any axis-parallel segment that lies completely inside of P; they prove that any simple rectilinear polygon with n vertices admits a decomposition with stabbing number $O(\log n)$, and they give an example of a simple rectilinear polygon

for which any decomposition has stabbing number $\Omega(\log n)$. They generalize their results to rectilinear polygons with rectilinear holes. Agarwal, Aronov, and Suri [2] investigate extremal properties of the stabbing number of triangulations in three dimensions, where the stabbed objects are simplices; see also Aronov and Fortune [5] for this problem. Shewchuk [24] shows that in d dimensions, a line can stab the interiors of $\Theta(n^{\lceil d/2 \rceil})$ Delaunay d-simplices. This implies, in particular, that a Delaunay triangulation in the plane may have linear stabbing number. More recently, Tóth [25] showed that for any subdivision of d-dimensional Euclidean space, $d \geq 2$, by n axis-aligned boxes, there is an axis-parallel line that stabs at least $\Omega(\log^{1/(d-1)} n)$ boxes, which is the best possible lower bound. Generalizations of the stabbing objects have also been considered: most notably, Chazelle and Welzl [6] describe extremal properties of stabbing spanning trees in d-dimensional space by hyperplanes; the analogous problem and a corresponding result for matchings is also discussed in [6].

This Paper.   The rest of this paper is organized as follows. After some basic definitions and notation in Section 2, we give details of our various hardness proofs in Section 3. In Section 4, we describe our LP-based approach for constructing bounds. Section 5 presents an iterated rounding technique for matching and spanning tree problems; we believe that the resulting algorithms yield constant-factor approximations. Final concluding thoughts and miscellaneous results and problems are presented in Section 6.

2   Preliminaries

Given a set L of line segments in the plane, the *stabbing number* of a line ℓ is the number of segments of L that are intersected by ℓ. The *stabbing number of L* is the maximum stabbing number over all lines ℓ; the *axis-parallel stabbing number of L* is the maximum stabbing number over all axis-parallel lines ℓ. In this paper, the set L will arise as the edges of a perfect matching, spanning tree, or triangulation of a given set P of n vertices in the plane, and our objective is to find such a structure of minimum stabbing number. Any reference to matching always means perfect matching. Therefore, when dealing with matchings, we assume that n be even, if necessary by omitting one of the vertices.

   We denote by $St\text{-}Mat(P)$ the minimum stabbing number among all matchings of P, by $St\text{-}Tre(P)$ the minimum stabbing number of all spanning trees of P, and by $St\text{-}\Delta(P)$ the minimum stabbing number of all triangulations of P. We use $St\text{-}Mat_2(P)$, $St\text{-}Tre_2(P)$, and $St\text{-}\Delta_2(P)$ for the minimum axis-parallel stabbing numbers.

   For a set $L = \{l_1, \ldots, l_n\}$ of line segments in the plane, the *crossing number* of a line ℓ is the number of connected components of $\bigcup_{i=1}^{n} l_i \cap \ell$. The *crossing number of L* is the maximum crossing number over all lines ℓ; just like for the stabbing number, the axis-parallel crossing number is defined for axis-parallel lines. For matchings, trees, and triangulations, we use the analogous abbreviations $Cr\text{-}Mat(P)$, $Cr\text{-}Tre(P)$, and $Cr\text{-}\Delta(P)$, and their subscripted counterparts $Cr\text{-}Mat_2(P)$, $Cr\text{-}Tre_2(P)$, and $Cr\text{-}\Delta_2(P)$ for the axis-parallel crossing numbers. Note that stabbing and crossing number coincide for planar matchings.

3   Complexity

In this section we prove $\mathcal{NP}$-hardness of computing the minimum stabbing number of matchings and computing the minimum crossing number of triangulations; for spanning trees the proofs are

analogous, and we only give a sketch of the proof. Our technique is rather general and should be applicable to other structures and variants as well.

3.1 Perfect Matchings

Theorem 1 *Deciding whether St-Mat$_2(P) \leq 5$ is strongly $\mathcal{NP}$-complete.*

Proof. Clearly, the problem is in $\mathcal{NP}$. We show completeness using a reduction from 3SAT [13]. Assume we have a Boolean expression denoted by $B(x_0, x_1, \ldots, x_{n-1})$ with n variables and k clauses of three literals each. We construct a set of vertices P that has a perfect matching M of stabbing number 5 if and only if the Boolean expression can be satisfied; in case of an unsatisfiable expression, a stabbing number of at least 6 cannot be avoided.

Consider the overall layout of P as shown in Figure 1. We make critical use of the collinearity of vertices, using up all of the available stabbing number of 5 in a particular direction. Thus we are able to construct "barriers" which avoid any interference between the different gadgets.

Figure 1: Overall layout of the construction for *St-Mat$_2(P)$*. Shown is the layout for the 3SAT instance $(x_0 \vee x_1 \vee x_3) \wedge (\overline{x_0} \vee x_2 \vee x_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_3})$, with a truth setting of $x_0 = true$, $x_1 = true$, $x_2 = false$, $x_3 = true$. Note that in full scale, the two gadgets representing the literal x_3 in clauses 0 and 1 have disjoint x-coordinates.

At the top of the layout are two groups of 10 vertices. The vertices in a group of 10 have the same y-coordinate. We call these two groups the *top rows*. The i-th vertex in the first top row has the same x-coordinate as the i-th vertex in the second top row. Below the top rows are n groups of 6 vertices. All vertices in a group of 6 have the same x-coordinate, as shown in the figure. The vertical

lines through these groups of 6 vertices separate the variables from each other and from other vertices left of the variables. We call each such group a *vertical barrier* gadget. The vertices in barrier i are sufficiently far below the vertices in barrier $i + 5$ to ensure that horizontal lines through the vertical barriers have stabbing number at most 5. The barriers lie between vertical lines through the last and second last vertices of the top rows. To the left of the top rows and below the vertical barriers are $k + 1$ groups of 10 vertices. Each vertex in a group of 10 has the same y-coordinate. We call each such group a *horizontal barrier* gadget. The vertices in barrier i are sufficiently far to the right of the vertices in barrier $i + 5$ to ensure that vertical lines through the horizontal barriers have stabbing number at most 5. The horizontal barriers are used to separate clauses from each other, to separate the clauses from the variables, and to separate variables from other vertices above the variables. Between and to the right of the top two horizontal barriers are groups of 8 vertices. Each vertex in a group of 8 has the same y-coordinate. We call each such group a *reducer* gadget of a variable. The vertices in reducer i are far enough to the right of the vertices in reducer $i + 1$ to ensure that vertical lines through the reducers have stabbing number at most 5.
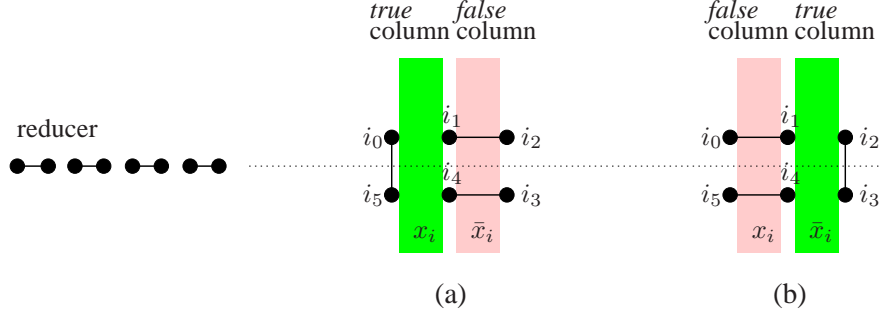
Figure 2: Variable x_i with $x_i = $ *true* in (a) and $x_i = $ *false* in (b).

Figure 2 shows a set of 6 vertices, numbered i_0 to i_5 in clockwise order, that represents a variable. These *variable* gadgets are separated by vertical barriers, and are placed below the vertical barriers, between the top two horizontal barriers and to the right of the reducers. Variables are placed such that the y-coordinate of the horizontal line through the reducer of a variable is in between the y-coordinate of vertices i_0 and i_5 of that variable. The strip formed by vertical lines between vertices i_0 and i_1 of variable x_i is called the x_i-*column* of the variable. The strip formed by vertical lines between vertices i_1 and i_2 of variable x_i is called the $\bar{x}_i$-*column* of the variable.
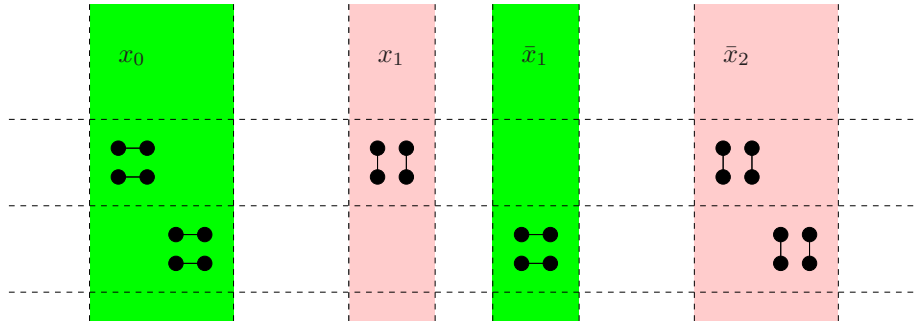

Figure 3: Clauses $(x_0 \vee x_1 \vee \bar{x}_2)$ and $(x_0 \vee \bar{x}_1 \vee \bar{x}_2)$ with $x_0 = x_2 = $ *true* and $x_1 = $ *false*.

The horizontal and vertical barriers create kn locations for *literal* gadgets. These are groups of 4 vertices representing the occurrence of a variable in a clause. Each group of 4 vertices forms an axis-parallel square. If a literal x_i appears in the clause c_j, we place a literal gadget in the x_i column of clause c_j. If a literal $\bar{x}_i$ appears in the clause c_j, we place a literal gadget in the $\bar{x}_i$ column of clause c_j. The three literal gadgets for a clause are put on the same horizontal lines. The literal in column x_i of clause c_j is to the left of the literal in column x_i of clause c_h for $j < h$. Similarly, the literal in column $\bar{x}_i$ of clause c_j is to the left of the literal in column $\bar{x}_i$ of clause c_h for $j < h$. Figure 3 shows the literals of two clauses.

We first assume that $B(x_0, x_1, \ldots, x_{n-1})$ is satisfiable, and show that P has a matching M of stabbing number 5. We connect vertex i to vertex $i+1$ in each of the two top rows for $i = 0, 2, 4, 6, 8$. We connect vertex i to vertex $i + 1$ in each vertical barrier for $i = 0, 2, 4$. We connect vertex i to vertex $i + 1$ in each horizontal barrier for $i = 0, 2, 4, 6, 8$. We connect vertex i to vertex $i + 1$ in each reducer for $i = 0, 2, 4, 6$.

If the variable x_i has the value *true*, we connect the pairs (i_0, i_5), (i_1, i_2), and (i_3, i_4) of the variable. If the variable x_i has the value *false*, we connect the pairs (i_0, i_1), (i_2, i_3), and (i_4, i_5). Notice that if x_i is *true*, then any vertical line in the x_i-column stabs 2 edges in the top rows of M, and a vertical line in the $\bar{x}_i$-column stabs 2 edges in the top row and 2 edges in the rectangle of the variable. If x_i is *false*, this situation is reversed. The column with vertical stabbing number 2 is called the *true*-column of the variable, the column with vertical stabbing number 4 is called the *false*-column. In each literal gadget representing the value *true* we connect the four vertices with two horizontal edges. In each literal gadget representing the value *false* we connect the four vertices with two vertical edges.

We can now verify that M has stabbing number 5. Any vertical line in the *true*-column of variable x_i stabs two edges in the top rows and at most two edges in a literal. Any vertical line in the *false* column stabs two edges in the top rows, two edges of the variable and at most one in a literal. Any horizontal line in a clause stabs at most three literal gadgets, one of which is set to *true*. So these lines stab at most 5 edges of M. It can easily be verified that all other horizontal and vertical lines stab at most 5 edges from M.

Conversely, we assume that P has a matching of stabbing number 5. We show that B is satisfiable. The matching used in this proof is illustrated in Figures 1, 2 and 3. Because the top rows contain 10 vertices, these vertices have to be connected to each other, otherwise the stabbing number of P exceeds 5. There are several ways to connect the sets of 10 vertices. If we connect vertex i to vertex $i + 1$ for $i = 0, 2, 4, 6, 8$ in each row, the number of edges stabbed by any horizontal or vertical line is minimized. Therefore we may assume without loss of generality that these edges are in the matching M. Collinear vertices are dealt with in a similar manner: If there are several ways to connect a set of collinear vertices, we will prefer connections that have minimal stabbing number for all stabbing lines.

Thus, we can connect vertex i to vertex $i + 1$ for $i = 0, 2, 4, 6, 8$ in each horizontal barrier gadget. For the same reason, we can connect vertex i to vertex $i + 1$ for $i = 0, 2, 4, 6$ in each reducer gadget. Because vertical lines through the vertical barrier gadgets stab two edges in the top rows, we can connect vertex i to vertex $i + 1$ for $i = 0, 2, 4$ in each vertical barrier gadget.

Now it is easy to see that no vertex in a variable or literal gadget for some variable x_i can be matched with any vertex not involved with representing the same variable x_i: otherwise, the edge would cross a line through a vertical barrier, which already crosses another five edges. Furthermore, such a vertex must be matched with vertices from the same gadget (either the same variable gadget or

the same literal gadget): otherwise, we get a violation at a line through a horizontal barrier gadget.

Now each reducer gadget contributes four to a horizontal stabbing number. Thus, we cannot connect the six vertices of a variable x_i by three vertical edges. Figure 2 shows the two remaining, essentially distinct matchings.

Of the three literals in each clause, one has to be set to true, otherwise there will be a horizontal stabber intersecting six edges. The true literal, say, x, must lie in a *true*-column of a variable, because vertical lines in this column have stabbing number two. Any other literal in this column can also be set to *true*. The literals $\bar{x}$ lie in the *false*-column of the same variable, and have to be set to false. So if a matching of stabbing number five exists, there is a truth assignment of the Boolean expression. □

Corollary 2 *There is no α-approximation algorithm for St-Mat$_2(P)$ with $\alpha < 6/5$; in particular there is no polynomial time approximation scheme (PTAS), unless $\mathcal{P} = \mathcal{NP}$.*

Corollary 3 *Computing St-Mat(P) is a weakly $\mathcal{NP}$-hard problem.*

Proof. We apply a perturbation technique, similar to the one in [10]. We start with the same basic construction as for the hardness proof for the axis-parallel case, and consider the grid formed by the coordinates of the resulting vertex set. This grid is modified such that the interpoint distances between the vertices of the same gadget are $\Theta(\varepsilon^{n^2+2})$ for the literal gadgets and $\Theta(\varepsilon^{n^2})$ for all other gadgets. Furthermore, the rest of the grid is perturbed by powers of ε, such that only axis-parallel lines can stab more than two gadgets; in particular, we increase the vertical distance between variable gadgets and the (narrower) literal gadgets by a sufficient amount, in order to make sure that no line through two literal gadgets for the same variable can intersect the corresponding variable gadget. Now it is easy to see that lines that are not axis-parallel can stab at most four line segments, leaving only axis-parallel lines as critical. □

3.2 Triangulations

Our basic proof technique is the same as for matchings. We first describe the construction of barrier gadgets, using the following terminology. A *horizontal line* is given by a set of vertices that are horizontally collinear. A *vertical line* is given by a set of vertically collinear vertices. A *row* consists of two horizontal lines, and the (empty) space between them. A *column* consists of two vertical lines, and the (empty) space between them.

Lemma 4 *Consider a row consisting of two horizontal lines l_a and l_b in P, having a and b vertices, respectively. If the combined number of edges on l_a and l_b is $a + b - i - 2$, then a horizontal stabber between l_a and l_b encounters at least $a + b + i - 2$ triangles in any triangulation of P and its crossing number is at least $a + b + i - 1$.*

Proof. Assume without loss of generality that l_a lies above l_b. Suppose there are $a - i_a - 1$ edges on the line l_a and $b - i_b - 1$ edges on the line l_b with $i_a + i_b = i$. For each edge (u, v) on l_a there is a triangle (u, v, w) where w lies either on or below l_b. Let A denote this set of triangles. Similarly, for each edge (u, v) on l_b there is a triangle (u, v, w) where w lies either on or above l_a. Let B denote this set of triangles. For each two neighboring vertices u and v on l_a for which there is no edge (u, v) there are triangles (u, u_0, u_1) and (v, v_0, v_1) such that u_0 and v_0 lie on or below l_b and u_1 and v_1 lie

above l_a. Let I_a denote this set of triangles. Also for each two neighboring vertices u and v on l_b for which there is no edge (u, v) there are triangles (u, u_0, u_1) and (v, v_0, v_1) such that u_0 and v_0 lie on or above l_a and u_1 and v_1 lie below l_b. Let I_b denote this set of triangles. It is not hard to verify that any two of the the four sets of triangles A, B, I_a and I_b have an empty intersection. A horizontal line l between l_a and l_b stabs every triangle in A, B, I_a and I_b. So l stabs at least $|A| + |B| + |I_a| + |I_b|$ $= (a - i_a - 1) + (b - i_b - 1) + 2i_a + 2i_b = a + b + i - 2$ triangles and $a + b + i - 1$ edges. $\square$

The lemma holds analogously for two vertical lines that form a column. When a row consists of two horizontal lines that have $Cr\text{-}\Delta_2(P) + 1$ vertices altogether, we call it *full* or fully triangulated. It follows from the lemma that all $Cr\text{-}\Delta_2(P) - 1$ edges on the lines l_a and l_b have to be present.

Theorem 5 *Finding $Cr\text{-}\Delta_2(P)$ is $\mathcal{NP}$-hard.*

Proof. Again we use a reduction from 3SAT, and the proof proceeds along the lines of the proof of Theorem 1. See Figure 4 for a schematic layout of a representing vertex set P for the 3SAT instance $B(x_0, x_1, x_2) = (x_0 \vee x_1 \vee \bar{x}_2) \wedge (x_0 \vee \bar{x}_1 \vee x_2) \wedge (\bar{x}_0 \vee \bar{x}_1 \vee \bar{x}_2)$. Figure 5 shows the structure of variable gadgets.
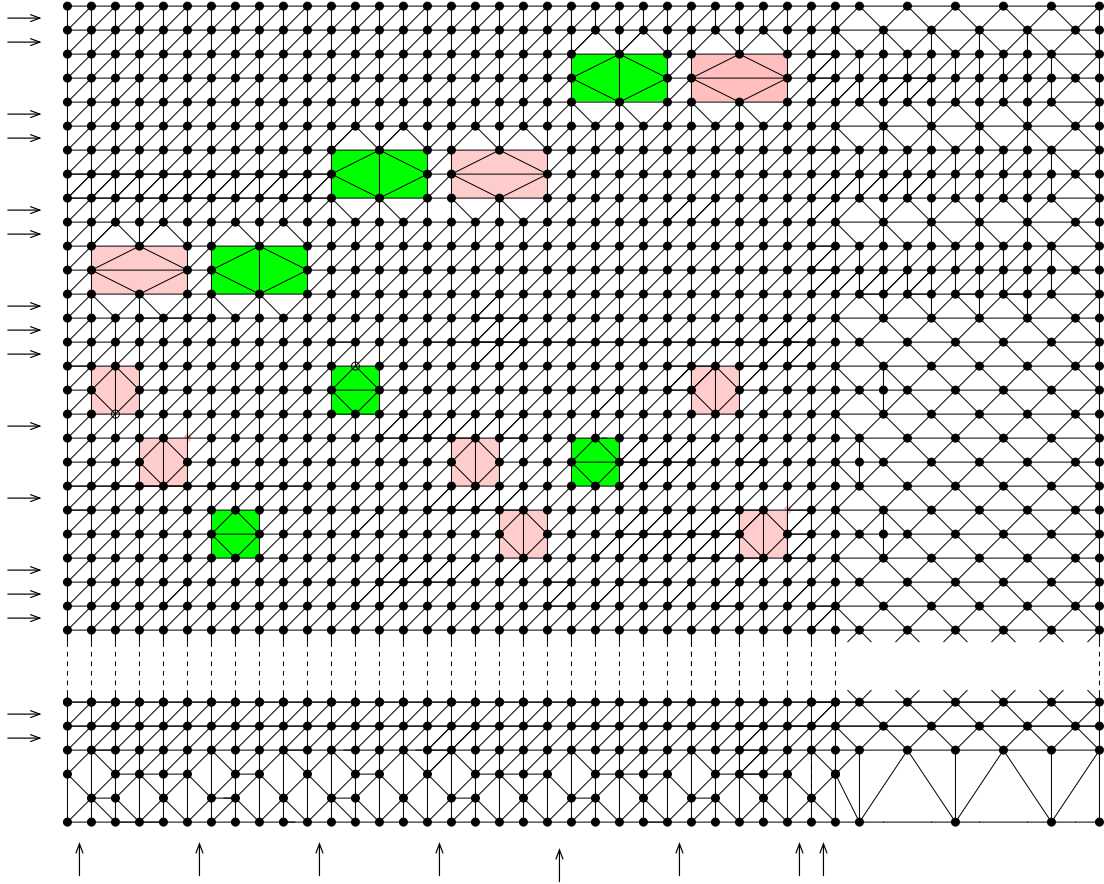

Figure 4: Overall layout for $Cr\text{-}\Delta_2(P)$. Clauses are $(x_0 \vee x_1 \vee \bar{x}_2)$, $(x_0 \vee \bar{x}_1 \vee x_2)$, and $(\bar{x}_0 \vee \bar{x}_1 \vee \bar{x}_2)$, with $x_0 = $ *false* and $x_1 = x_2 = $ *true*. Arrows indicate full rows and columns, light or dark shading indicates true or false variables and literals as before.

For a given Boolean expression $B(x_0, x_1, \ldots, x_{n-1})$ with n variables and k clauses of three literals each we construct a set P of vertices. We show that there is a value K such that B is satisfiable only if Cr-$\Delta_2(P) = 2K - 1$; if B cannot be satisfied, Cr-$\Delta_2(P)$ is at least $2K$.

In Figure 4 we have $K = 39$ and a grid of vertices with some well-defined holes. The maximum number of vertices in a horizontal or vertical line is K, and many lines have exactly K vertices. By Lemma 4 a full row or column in this setting has exactly $2K - 2$ triangles. Gadgets are separated by full rows and columns.
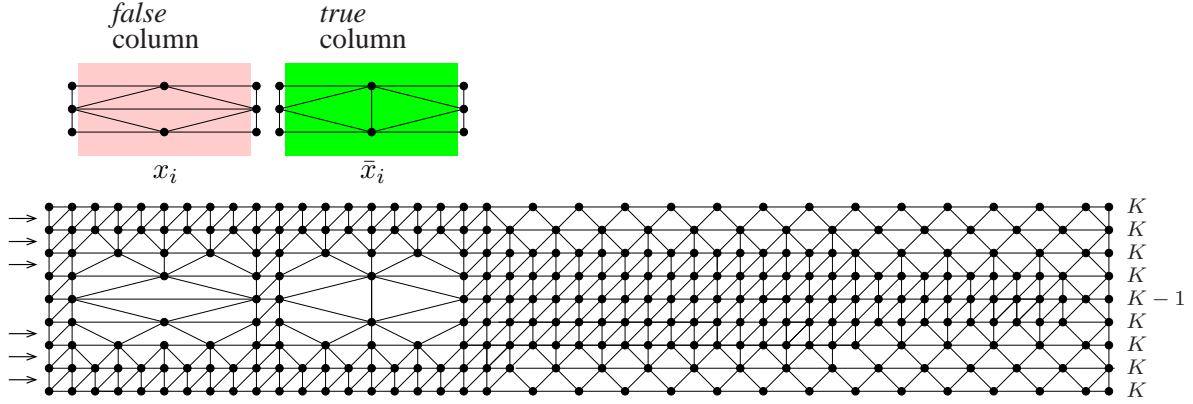
Figure 5: A variable gadget and how it is embedded in a grid of vertices. Arrows indicate full rows.

Figure 5 shows two horizontally aligned rectangles of eight vertices each that together represent a variable x_i. We call the strip formed by vertical lines that stab the left rectangle the x_i-*column*, and the strip formed by vertical lines that stab the right rectangle the $\bar{x}_i$-*column* of the variable. The gadget works essentially the same way as that in Figure 2. Each rectangle has full rows and columns as neighbors. We indicate how this can be achieved in horizontal direction in Figure 5. By Lemma 4 we conclude that all edges on the boundary of the convex hull of each rectangle are present in any triangulation of minimal crossing number. The horizontal lines that contain the top and bottom lines of the two rectangles, respectively, contain K vertices each; the horizontal line that passes through the middle of the rectangles contains $K - 1$ vertices. Therefore Lemma 4 shows that exactly one edge along this horizontal line in the middle of the rectangle may be missing. We call the strip that is spanned by all vertical lines that stab the rectangle with the missing horizontal edge the *true*-column of the variable x_i. The strip that is spanned by all vertical lines that stab the rectangle for which the middle horizontal edge is present is the *false*-column of the variable x_i. As shown in Figure 4, any satisfying literal adds exactly one less to the vertical crossing number than an unsatisfied one. In the overall layout, variable x_i is placed below and to the left of variable x_j for $i < j$ in such a way that variables are vertically separated from one another by a full row, and horizontally separated from one another by a full column.

Each literal is represented by a square with eight vertices on its boundary. We make the width of rectangles of the variables equal to a power of two, and wide enough to accommodate the necessary number of literals. That is, each rectangle of a variable is of a width at most four times the number of occurrences of the most frequent literal in B. Figure 5 gives a hint at how this widening of a rectangle is done. Notice that if the most frequent literal in B occurs t times, there are $\Theta(\log t)$ rows above and below the variable so that in the top and bottom lines of these groups of rows the vertices above and below the variables are a distance of one apart. The three literals of a clause c_j are horizontally

10

aligned, and the two rows that are spanned by them are called the clause c_j. Clauses are separated from each other by full rows. If a literal x_i appears in the clause c_j, we place a literal gadget in the x_i-column of the clause gadget c_j. If a literal $\bar{x}_i$ appears in the clause c_j, we place a literal gadget in the $\bar{x}_i$-column of the clause gadget c_j. The literal in column x_i of clause c_j is to the left of the literal in column x_i of clause c_h for $j < h$. Similarly, the literal in column $\bar{x}_i$ of clause c_j is to the left of the literal in column $\bar{x}_i$ of clause c_h for $j < h$. So no vertical line stabs the interior of more than one literal.

By adding vertices to the right of the literals we ensure that a horizontal line through the top or bottom row of the three literals of a clause has exactly K vertices, and a horizontal line through the middle horizontal line of the three literals of a clause has exactly $K - 2$ vertices. So two edges along these middle lines may be missing in a triangulation of minimum crossing number, but no more than two. As we will argue later, these edges will be missing in the interior of at most two of the literals. We call the missing of the horizontal middle edge in a literal the *false* setting of the literal, and the presence of this edge within a literal the *true* setting of the literal.

Because the rows above and below a variable are full, we can assume that they are triangulated as shown in Figure 5, because any other triangulation would result in strictly higher vertical crossing numbers. So the *true* column of a variable has a vertical crossing number that is one less than the crossing number of a *false* column. In adding vertices at the bottom of the clauses we ensure the following vertical vertex counts. First of all, the columns neighboring the variables have to be full; in particular, vertical lines that stab the left or the right vertices of a variable rectangle have K vertices each. All other vertical lines through a variable should get a number of vertices so that if the corresponding column of the variable is set to *false*, all remaining edges on these vertical lines have to be present. This implies that if the column is set to *true*, we can have one missing edge in the vertical line to the right of the left boundary of the rectangle, and if this edge is missing, there cannot be a missing edge in the next vertical line, one edge missing in the next line, etc.

Let $B(x_0, x_1, \ldots, x_{n-1})$ be satisfiable. We show that P has a triangulation of crossing number $2K - 1$ that is minimum by Lemma 4. All full rows and full columns are fully triangulated. If variable x_i has the value *true*, we triangulate the interior of the two rectangles of variable x_i in such a way that the x_i-column becomes this variable's *true*-column, and the $\bar{x}_i$-column becomes this variable's *false*-column. The triangulation of the interior of the rectangles is reversed when variable x_i has the value *false*. We set each literal that represents the value *true* to its *true* setting, and set each literal that represents the value *false* to its *false* setting. The triangulation can be completed arbitrarily.

We can now convince ourselves that such a triangulation of P indeed attains $Cr\text{-}\Delta_2(P) = 2K - 1$. No fully triangulated row or column has a crossing number larger than $2K - 1$. Because exactly one edge is missing in the horizontal middle line of each variable, Lemma 4 implies that the crossing numbers of the two rows of a variable are both equal to $(2K - 1) + 1 - 1 = 2K - 1$. In each clause there is at least one literal in its *true* setting. Therefore, we can afford two extra triangles caused by the *false* setting of the other two literals in the clause, and no row of a clause intersects more than $2K - 2$ triangles. Finally, we may have one edge missing on the vertical lines passing trough the middle of a literal only in a *true*-column. This condition holds by definition of the setting of the literals according to the truth value of the variables. The "arbitrary completion" of the triangulation only happens in the lower right of the construction. In this corner, vertical and horizontal lines have a low vertex count (except for the boundary), and the allowed crossing number is not exceeded.

For seeing the converse, assume that there is a triangulation of P that has crossing number $2K - 1$. We show that B is satisfiable. Because a full row or column can be triangulated in such a way that

the crossing number of $2K - 1$ is not exceeded, we only have to take care of the rows and columns in which we have a degree of freedom, and where the vertex count is critical by Lemma 4. Because both the top row and the bottom row of a clause have $2K - 2$ vertices, we can afford at most two literals of each clause set to *false*. One literal in each clause has to be set to *true*. The *true* literal has to be in a *true*-column of a variable, for otherwise the vertical crossing number would exceed $2K - 1$. Any other literal in this column can also be set to *true*. The horizontal vertex count of the horizontal lines of the variable forces the second column of this variable to be a *false*-column. In order not to exceed the allowed crossing number in vertical direction, all literals in this column have to be set to *false*. As one easily checks, this yields a consistent setting of the variables, and B is satisfiable.

Finally, the size of our construction is indeed polynomial: let n and c be the number of variables and clauses of $B(x_0, x_1, \ldots, x_{n-1})$. Let t be the number of times the most frequent literal in used in B. A rectangle that represents a variable has width at most $4t$ and requires $\Theta(\log t)$ rows. The number of rows used by the clauses is $\Theta(c)$. In order to achieve the correct vertex count in each line we may have to add vertices to the right and below, which requires at most $\Theta(t)$ additional rows and columns. Therefore K is polynomial in n, c and t. □

3.3 Spanning Trees

The basic construction for showing hardness of finding a spanning tree of minimum stabbing number is similar to the one for matchings. As before, we use barriers to restrict possible connections: we make use of the arrangement shown in Figure 6, which works as a barrier gadget because of the following lemma.

Lemma 6 *Consider three parallel lines, ℓ_1, ℓ_2, ℓ_3, with a set S_i of k vertices on line ℓ_i, $i = 1, 2, 3$; let $S = S_1 \cup S_2 \cup S_3$. Consider $P \supset S$ and a spanning tree T of P with stabbing number $k + 1$. Then no edge of T crosses the strip spanned by the three lines.*
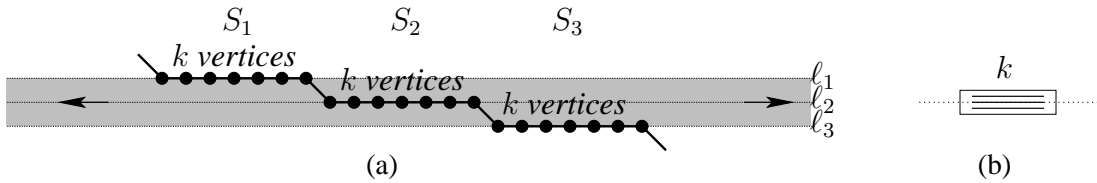

Figure 6: A horizontal barrier gadget, consisting of $3k$ vertices: (a) In a spanning tree of stabbing number $k + 1$, no edge may cross the shaded region. (b) Symbol for the barrier gadget; the dotted line indicates the blocked strip.

Proof.   See Figure 6. Consider a spanning tree T of P, with $v \in P \setminus S$ lying outside of the strip. Orient all edges of T towards v. Each vertex in S must have outdegree 1, meaning that there are k outgoing edges for each of S_1, S_2, S_3, contributing k to the stabbing numbers along ℓ_1, ℓ_2, ℓ_3. One of the outgoing edges of S_2 must intersect ℓ_1 or ℓ_3 in order to connect S_2 to the rest of the graph; thus, one of those two lines stabs $k + 1$ edges, implying the claim. □

Our variable gadgets look as in Figure 7; shown is the gadget for variable x_i; note that the gadgets for $x_0, \ldots, x_{i-1}$ are left and below the box spanned by the gadget, while the gadgets for $x_{i+1}, \ldots, x_{n-1}$ are above and to the right of the spanning box. The bold squares below the gadget indicate the position of literal gadgets, which wil be discussed further down.

12

Also note the use of vertical barrier gadgets: a number of i next to it indicates a gadget consisting of $3i$ vertices, which alread requires a crossing number of $i + 1$; thus, only $k - i$ additional edges in a spanning tree may cross the dotted line induced by such a gadget. The arrows pointing down from the bottom indicate a number of literal gadgets, consisting of 2×2 arrangements of vertices. See Figure 8 for the resulting overall arrangement.

Lemma 7 *Let S be the arrangement of vertices shown in Figure 7, with barrier gadgets placed and sized as indicated, and let $P \supseteq S$. Let P be constructed as shown in Figure 8. Then any spanning tree of P that has stabbing number at most $k + 1$ must use at least one of the two edges at the bottom of the arrangement, labeled $e_t^{i,1}$ (for* true*) or $e_f^{i,1}$ (for* false*.)*
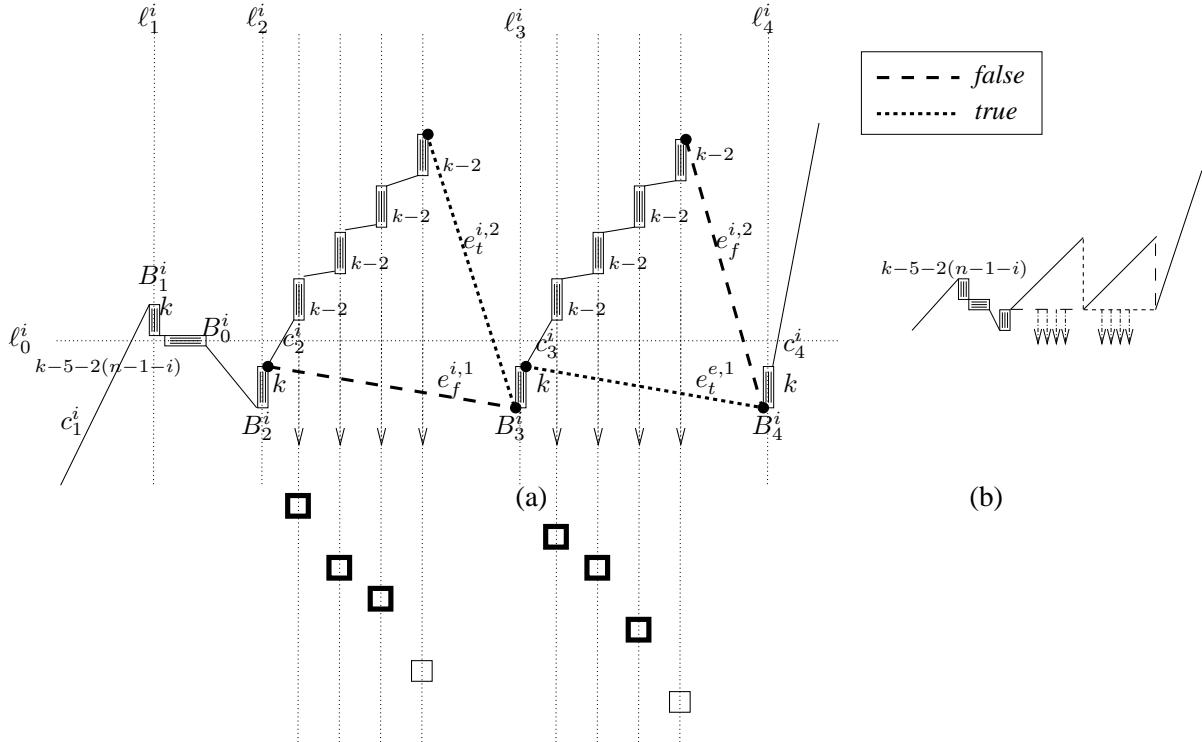
Figure 7: A variable gadget for variable x_i. (a) In a spanning tree of stabbing number $k + 1$, the *true* or the *false* setting is chosen. (b) Symbol for the variable gadget.

Proof. Assume there is a spanning tree of stabbing number at most $k+1$. Consider the barrier gadgets labeled B_1^i, B_2^i, B_3^i, B_4^i, and the corresponding lines, ℓ_1^i, ℓ_2^i, ℓ_3^i, ℓ_4^i. By the previous lemma, no edge can cross one of those lines. Therefore, the literal boxes below each clause must be connected within the vertical strips bounded by ℓ_2^i and ℓ_3^i on one hand, or ℓ_3^i and ℓ_4^i, on the other hand. This requires at least one edge within each of the two strips to cross the line ℓ_0^i. Moreover, the lines ℓ_1^i and ℓ_4^i must not be crossed, implying that the variable gadgets are connected to their neighbors at barriers B_1^i and B_4^i. These connections between neighboring variable gadgets form a stair-like chain of variable gadgets, as shown in Figure 8: connections are in increasing x- and y-order, which correspond to increasing variable indices; we call this the *exterior stair*. Similarly, the barrier gadgets associated with each of the two truth settings of a variable form one stair-like chain each, as shown in Figure 7; we call these the two *interior stairs*.

13

Now consider the horizontal barrier B_0^i, consisting of three groups of $k - 5 - 2(n - 1 - i)$ vertices each. By the previous arguments, the line ℓ_0 has to cross all of the edges connecting the true and false literal boxes of the $n - 1 - i$ variables with higher indices, i.e., cross $2(n - 1 - i)$ edges. Furthermore in variable x_i there are four other edges that connect vertices above line l_0^i to vertices below l_0^i and so are crossed by l_0^i. For example these edges could be the ones labeled $c_1^i, c_2^i, c_3^i, c_4^i$ in the figure. This allows only one of the edges $e_t^{i,2}$ and $e_f^{i,2}$ to be used for connecting the two interior stairs with each other and with the exterior stair; thus, at least one of the edges $e_f^{i,1}$ and $e_t^{i,1}$ must be used, proving the claim. $\square$

Making use of the above gadgets, we get

Theorem 8 *It is $\mathcal{NP}$-hard to determine St-Tre$_2(P)$.*

Proof. The basic idea for the construction is similar to the one used in the previous sections, making use of Lemmas 6 and 7. The use of gadgets and the overall layout of the construction are shown in Figure 8.
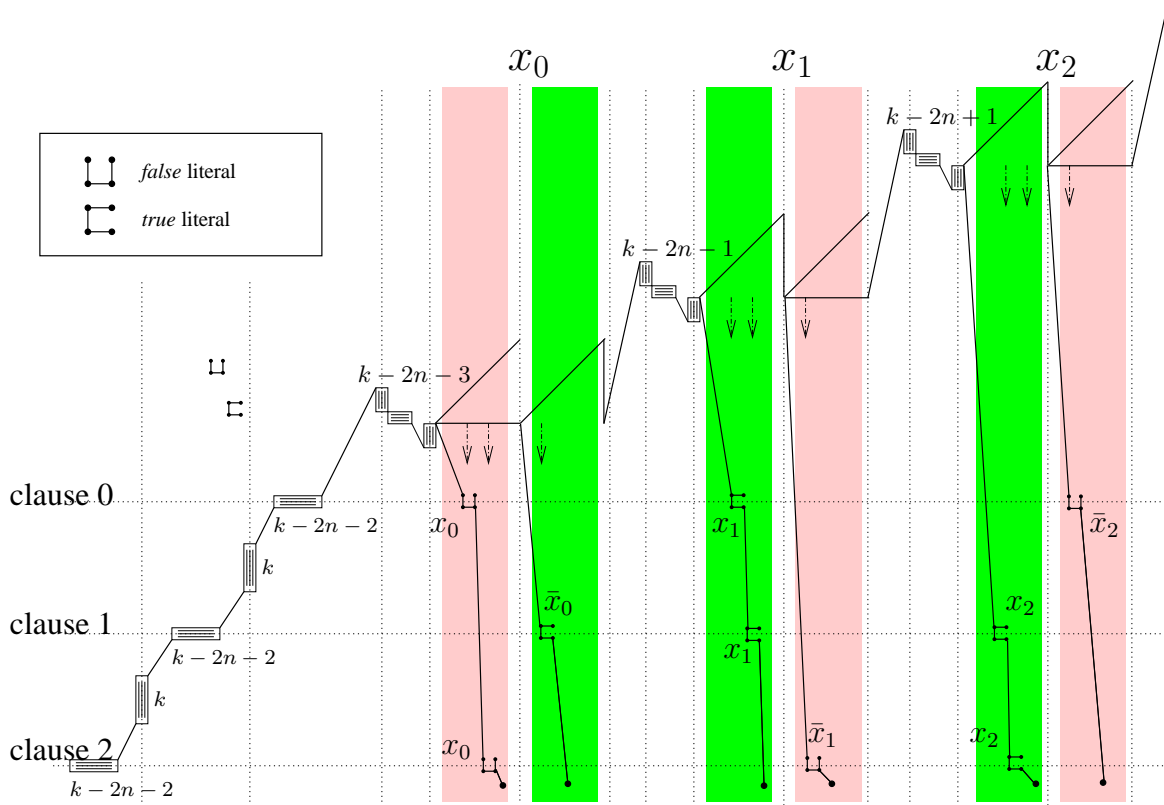

Figure 8: The overall layout for the hardness proof for spanning trees. There is a total of n variables; k is a sufficiently large number. In question is the existence of a spanning tree with stabbing number $k+1$. Shown is the representation of the 3SAT instance $(x_0 \vee x_1 \vee \bar{x}_2) \wedge (\bar{x}_0 \vee x_1 \vee x_2) \wedge (x_0 \vee \bar{x}_1 \vee x_2)$, for $n = 3$, with $x_0 = $ *false* and $x_1 = x_2 = $ *true*.

Given a Boolean expression $B(x_0, x_1, \ldots, x_{n-1})$, we can find a vertex set that has a spanning tree of a stabbing number $k + 1$, if and only if B is satisfiable. Consider the vertex set as given in

Figure 8 with $k = 3n$. If $B(x_0, x_1, \ldots, x_{n-1})$ has a satisfying truth assignment, we first connect the vertex on the left side of the drawing into one long path. This path contains all horizontal and vertical barrier gadgets as shown in Figure 8. We connect the variable and literal gadgets according to their values in B. In each *true*- and *false*-column, there is a vertex lower than all horizontal barriers, and to the right of all literals in that column. We connect this vertex to the right lowest vertex of the lowest literal in the same column. Because each clause has at least one true literal, the horizontal stabbing number of a stabber through a clause is at most k, because the horizontal barrier on the left contributes $k - 2n - 1$, the variable(s) set to true and the $n - 3$ variables that do not appear in the clause contribute two each, and the variable(s) set to false contribute three each, which is at most $k - 2n - 1 + 2(n - 2) + 6 = k + 1$. A horizontal stabber through the horizontal barrier in the i-th variable gadget stabs $k - 5 - 2(n - 1 - i) + 1$ edges in the barrier, one more edge to the left of the barrier, four edges of the variable to the right of the gadget and two more for each of the $n - i$ variables to its right, for a total of $k - 5 - 2(n - 1 - i) + 1 + 5 + 2(n - 1 - i) = k + 1$. A vertical stabber through literals stabs at most $k - 1$ edges in the vertical barriers, plus two more, either one from a false literal and one in the variable gadget, or two edges in the true literals. So we have a spanning tree of stabbing number $k + 1$.

Conversely, assume that the vertex set has a spanning tree of stabbing number $k + 1$. As we showed in the proof of Lemma 7, the literal gadgets can only be connected within their respective strips, forcing at least $2n$ stabbed edges. Furthermore, each *false* literal (being connected in a "u"-like fashion) causes an additional stabbed edge, while a *true* literal (connected in a "c"-like fashion) does not cause any additional stabbings. Thus, each clause must contain at least one *true* literal. Because of Lemma 7, at least one of the edges e_f^1 and e_t^1 must be present, guaranteeing that only the negated or only the unnegated literals for each variable can be connected in a "c"-like fashion, i.e., forcing a feasible setting of the variables. Thus, we get a truth setting of the variables that satisfies B. □

This immediately implies

Corollary 9 *It is $\mathcal{NP}$-hard to determine St-Tre(P).*

Proof. The argument is similar to the one in Corollary 3 for matchings: use the construction of Theorem 8, for which the criticality of certain axis-parallel lines requires satisfying a 3SAT instance in order to achieve low stabbing number. Scale down the bounding boxes for all gadgets, with literal gadgets ending up in appropriately smaller bounding boxes. Then perturb the position of the gadgets, shifting all vertices in the same gadget by the same amount, such that no line can intersect the bounding boxes of any three gadgets, again inserting a sufficient vertical distance for excluding a line that stabs a variable gadget and two of its literal gadgets. This leaves only the axis-parallel lines to be critical, implying the same combinatorial behavior as in the axis-parallel case. □

The hardness proof for minimizing the crossing number has the same structure as the one for stabbing number. Instead of the barrier gadget implied by Lemma 6, we use a slightly different one, as shown in Figure 9.

Lemma 10 *Let S be the $k \times ((k-1)^2 + k)$ arrangement of vertices shown in Figure 9, and let $P \supseteq S$. If P has a spanning tree T with crossing number k, then no edge of T connecting two vertices outside of the arrangement crosses the horizontal strip spanned by the arrangement.*

Proof. Suppose there is a crossing edge. Consider the $(k-1)^2 + k$ horizontal lines that pass through k vertices of the vertex arrangement. Because of the crossing edge, the intersection of each line with
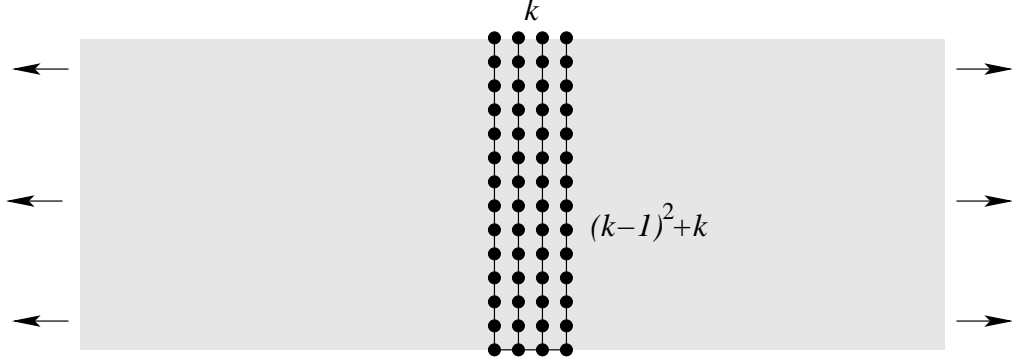
Figure 9: A barrier gadget for showing hardness of minimizing the crossing number of a spanning tree.

the edges and vertices within each corresponding horizontal set cannot consist of more than $k - 1$ connected components, so this intersection must contain at least one horizontal edge within the set; thus, a total of at least $(k - 1)^2 + k$ horizontal edges within the arrangement cannot be avoided. Now consider the k vertical lines that pass through $(k - 1)^2 + k$ vertices of the arrangement; these lines subdivide the plane into $k+1$ vertical strips, $k-1$ of which have width 1. Any of the at least $(k-1)^2+k$ horizontal edges within the arrangement must cross at least one of the intermediate strips; thus, the average number of horizontal edges per bounded strip is at least $\frac{(k-1)^2+k}{k-1} = (k - 1) + \frac{k}{k-1} > k$. By the pigeonhole principle, this implies that there must be a bounded vertical strip that is crossed by more than k edges, a contradiction to our assumption that the crossing number is at most k. □

As the figure shows, there is a feasible subtree of the gadget, as long as no edge crosses the indicated strip. This allows us to use the arrangement as a barrier gadget. As for the rest of the construction for the proof of Theorem 8, stabbing and crossing numbers coincide, this immediately implies the following.

Theorem 11 *It is $\mathcal{NP}$-hard to determine Cr-Tre$_2(P)$.*

From this it is easy to derive the following, again using a perturbation argument.

Corollary 12 *It is $\mathcal{NP}$-hard to determine Cr-Tre(P).*

Proof. The argument is similar to the one in Corollary 9: use perturbation to ensure that only axis-parallel lines can stab more than two gadgets. The only additional difficulty that has to be overcome is the fact that a diagonal line through one of the barrier gadgets in Figure 9 may have crossing number k: in principle, crossing number $2k$ could arise from stabbing two such gadgets in a non-axis-parallel manner. However, scaling the bounding boxes of the barrier gadgets in a way that the vertical dimension is much smaller than the horizontal dimension (say, by a factor of $O(\varepsilon^{n^3})$) makes sure that only lines of slope within range $[-\varepsilon^{n^3}, \varepsilon^{n^3}]$ achieve crossing number k for one gadget. Thus, only almost horizonal lines are of concern; it is easy to see that the above perturbation guarantees that no such line intersects more than one barrier gadget. Therefore, only axis-parallel lines can be critical, and the hardness proof remains valid.

4 Integer Linear Programs for Minimum Stabbing Number

In view of the negative complexity results for our problems there are two major directions to proceed: providing (good) lower bounds on the minimum stabbing number in order to obtain approximation algorithms; and insisting on optimality despite $\mathcal{NP}$-hardness. Our (integer) linear programming approach is an elegant way to combine both issues. We deal with them in the next two sections.

4.1 Perfect Matchings

In combinatorial optimization, P corresponds to the vertex set of a straight-line embedded complete graph $G = (P, E)$; then a matching M can be represented by its edge incidence vector $x \in \{0, 1\}^E$, where $x_{ij} = 1$ if $ij \in M$, and $x_{ij} = 0$ otherwise. Using these variables we are able to state an integer linear program for finding a perfect matching of minimum stabbing number. For $S \subseteq P$, denote by $\delta(S) = \{ij \in E \mid i \in S, j \notin S\}$ the *cut* induced by S.

$$\text{minimize} \quad k \tag{1}$$

$$\text{s.t.} \quad \sum_{ij \in \delta(\{i\})} x_{ij} = 1 \qquad \forall\, i \in P \tag{2}$$

$$\sum_{ij \in \delta(S)} x_{ij} \geq 1 \qquad \forall\, \emptyset \neq S \subset P,\ |S|\ \text{odd} \tag{3}$$

$$\sum_{ij : ij \cap \ell(d) \neq \emptyset} x_{ij} \leq k \qquad \forall\ \text{stabbing line}\ \ell(d)\ \text{in direction}\ d \tag{4}$$

$$x_{ij} \in \{0, 1\} \qquad \forall\, ij \in E \tag{5}$$

We obtain the associated linear programming (LP) relaxation by replacing (5) by

$$x_{ij} \geq 0 \tag{5'}$$

The inequalities (2) and (3) are necessarily satisfied for any perfect matching given by x, where the *blossom constraints* (3) ensure that any subset of vertices of odd cardinality has at least one edge to the outside. In his seminal paper Edmonds [9] showed that—together with (5')—these inequalities already constitute the complete description of the perfect matching polytope, that is, its extreme vertices exactly correspond to the incidence vectors of perfect matchings in G, i.e., the LP relaxation is integral.

To this description we add the stabbing constraints (4), for which we count the number of intersections of matching edges with any given line; this number is bounded by the variable k, and k is minimized. We have to choose this way of modeling because of our min-max objective. An optimal solution x to the integer program (1)–(5) represents a matching with stabbing number exactly *St-Mat*(P). In a pure integer programming description, the blossom constraints are implied; however, Figure 10 shows that the introduction of stabbing constraints yields a polytope that is no longer integral, which is to be expected for an $\mathcal{NP}$-hard problem. That is why we consider the LP relaxation, for which the use of blossom constraints yields a considerably tighter set of solutions. We will make use of this fact in the following section; here we comment on the complexity of the given LP and its solution.

We can limit the number of stabbing constraints (4): for the axis-parallel stabbing number, we can assume without loss of generality that a critical stabbing line runs through a vertex, so we get
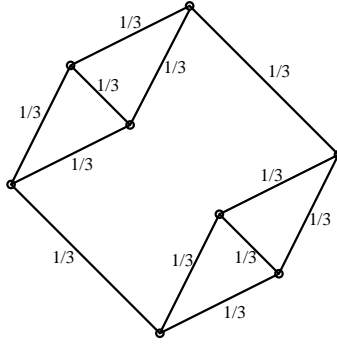
Figure 10: An optimal fractional solution of value $4/3$ with maximum edge weight $1/3$.

at most $O(n)$ restrictions for the axis-parallel stabbing number; for the general stabbing number, we may assume that a critical stabbing line runs through two vertices, so we can limit ourselves to $O(n^2)$ restrictions. On the other hand, we have exponentially many blossom inequalities; however, it is well known that blossom inequalities can be separated in polynomial time [22], i.e., one can check in polynomial time whether a given x violates some blossom inequality, and if so, identify such an inequality. This polynomial-time *separation* allows us to solve the linear programming relaxation (1)–(5′) in strongly polynomial time [23, Thm. 5.11] by means of the ellipsoid method [14, Thm. 66.5]. An optimal solution x will in general be fractional, and we speak of *fractional stabbing number* in this context. It is a lower bound on *St-Mat(P)*.

4.2 Spanning Trees

There are several polynomial-size LP formulations for spanning trees, see e.g., [19]. However, similar to matchings, we choose an exponential-size integer program that is again based on cut constraints. We directly state the LP relaxation.

$$\text{minimize} \qquad k \tag{6}$$

$$\text{s.t.} \qquad \sum_{ij \in E} x_{ij} = n - 1 \tag{7}$$

$$\sum_{ij \in \delta(S)} x_{ij} \geq 1 \qquad \forall \emptyset \neq S \subset P \tag{8}$$

$$\sum_{ij : ij \cap \ell(d) \neq \emptyset} x_{ij} \leq k \qquad \forall \text{ stabbing line } \ell(d) \text{ in direction } d \tag{9}$$

$$x \geq 0 \tag{10}$$

Equation (7) ensures the right number of edges in a tree solution, and connectivity is given by the cut constraints (8). Just like blossom constraints for matching problems, separation over these constraints can be done in polynomial time by means of a minimum cut routine, so this LP can also be solved in strongly polynomial time.

5 Iterated Rounding

The above LP relaxations provide lower bounds for optimal solutions, but no upper bounds in the form of feasible solutions for our stabbing problems (except if we solve the integer programs, which may take exponential time). Therefore, our next objective is to find, in polynomial time, an integer solution that is not too far from an optimal one.

For this purpose we consider the *support graph* of a (fractional) solution x: it consists of all edges e that have a strictly positive value $x_e > 0$. If we could be assured that in any optimal LP solution, all edges e in the support graph solution were b-heavy, i.e., $x_e \geq b$ for some constant $b > 0$, then we could consider rounding up all positive edge weights to one in order to get a $1/b$-approximation. Unfortunately, Figure 11 gives an indication that there may not be such a lower bound.

An alternative to simple LP rounding is offered by the iterated rounding technique introduced by Jain [17] for generalized Steiner network problems: at each step, it suffices to identify one heavy edge, round it up to one, fix it, and re-solve the remaining LP. The key ingredient is that the modified LP also has a heavy edge, so the process can be iterated, hence the name: *iterated rounding*.

A crucial step in Jain's seminal work was the fact that for so-called generalized Steiner networks, $1/2$ is a valid lower bound for the value of the heaviest edge of fractional weight. A sufficient condition for the validity of this bound that is purely combinatorial is that all constraints in the LP relaxation are cut constraints, i.e., of type $\sum_{e \in \delta(S)} x_e \geq b(S)$.

Unfortunately, stabbing constraints do not fall into the category of cut constraints, and it can be seen from Figure 10 that Jain's lower bound does not hold for stabbing problems. This makes it necessary to establish a separate lower bound on the value of the heaviest fractional edge. We will do this by exploiting the underlying geometric nature of our optimization problem, and establish the fact that there always is an optimal LP solution with planar support graph. Planarity is proven by shifting weight from a crossing pair of edges to a non-crossing one; for matchings (Lemma 13) this requires some extra care because of the blossom inequalities. The proof for spanning trees (Lemma 14) is almost completely analogous.

Lemma 13 *For any even set of vertices in the plane, there is a fractional perfect matching x of minimum stabbing number, such that the support graph of x is planar. Such a fractional matching can be found in polynomial time.*

Proof. The set of all LP solutions is bounded and nonempty, so the set of all optimal solutions is nonempty and compact. From this set, consider a solution x that minimizes the total stabbing number, i.e., the sum of stabbing numbers over all combinatorially different lines. We claim that the support graph of x cannot contain any crossing pair of edges. Refer to Figure 12.

Suppose there were $e_{13} := \{v_1, v_3\}$, $e_{24} := \{v_2, v_4\}$ with $x_{e_{13}} > 0$ and $x_{e_{24}} > 0$, such that e_{13} and e_{24} cross. We will argue that this implies the existence of an alternative solution of the same maximum stabbing number, but infinitesimally smaller total stabbing number, i.e., a contradiction to our assumptions on optimality.

Consider $e_{12} := \{v_1, v_2\}$, $e_{34} := \{v_3, v_4\}$, $e_{14} := \{v_1, v_4\}$, $e_{23} := \{v_2, v_3\}$. Among all LP constraints (1)–(5′), let $s(x)$ be the smallest positive slack (i.e., difference between left-hand and right-hand side), and choose $0 < \varepsilon < s(x)$. As $\sum_{e \in \delta(v_i)} x_e = 1$ and $x_{e_{13}} > \varepsilon$, $x_{e_{24}} > \varepsilon$, we have $x_{e_{12}} < 1 - \varepsilon$, $x_{e_{34}} < 1 - \varepsilon$, $x_{e_{14}} < 1 - \varepsilon$, $x_{e_{23}} < 1 - \varepsilon$.

Consider two possible alternative solutions arising from shifting an ε of weight from e_{13} and e_{24}.
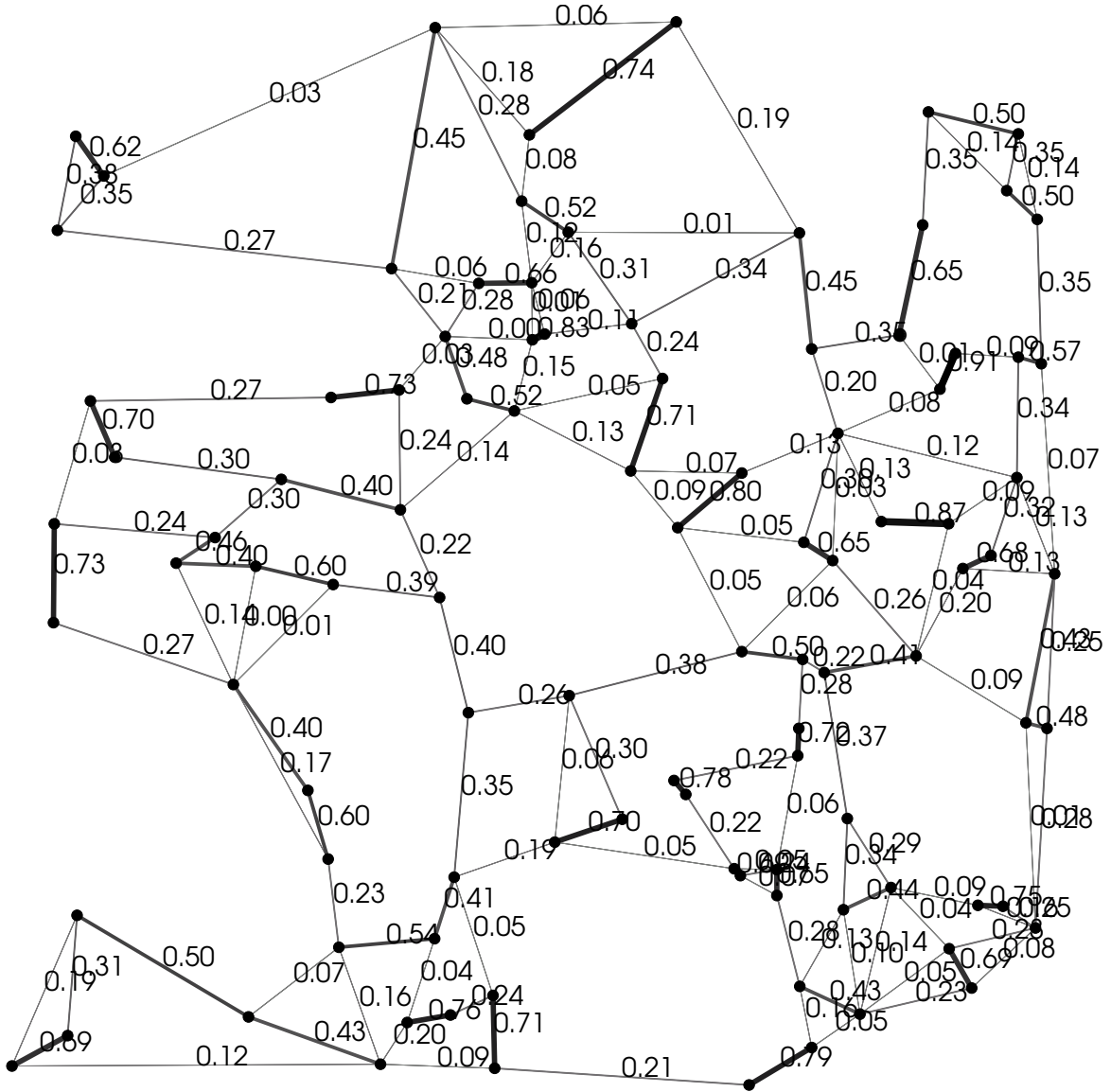
19

Figure 11: Finding a fractional matching of minimum stabbing number: shown is an optimal LP solution for an instance of 100 random vertices; see [12] for technical details and results for other instances.
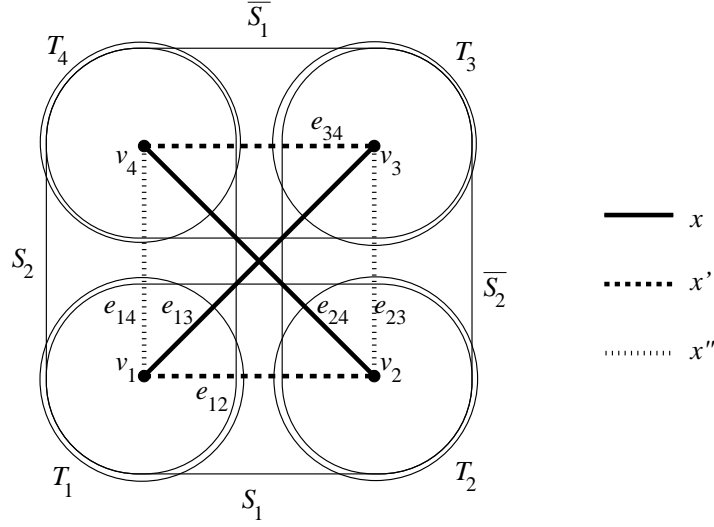
Figure 12: Improving a fractional solution with crossing edges while preserving all blossom inequalities.

Let x' and x'' be defined by

$$x'_e := \begin{cases} x_e - \varepsilon & \text{for} & e \in \{e_{13}, e_{24}\}, \\ x_e + \varepsilon & \text{for} & e \in \{e_{12}, e_{34}\} \\ x_e & \text{for} & \text{all other } e, \end{cases} \tag{11}$$

and

$$x''_e := \begin{cases} x_e - \varepsilon & \text{for} & e \in \{e_{13}, e_{24}\}, \\ x_e + \varepsilon & \text{for} & e \in \{e_{14}, e_{23}\}, \\ x_e & \text{for} & \text{all other } e. \end{cases} \tag{12}$$

By convexity, both x' and x'' satisfy all stabbing constraints that are valid for x. Furthermore, it is easy to see that both x' and x'' have smaller total stabbing number than x. Thus, it suffices to argue that at least one of these solutions satisfies all blossom inequalities.

Assume that for each of the two alternative solutions, a blossom inequality is violated; by our assumption on ε, this means that there are two odd sets, $S_1 \subset P$ and $S_2 \subset P$, such that $\sum_{e \in \delta(S_1)} x'(e) < 1$ and $\sum_{e \in \delta(S_2)} x''(e) < 1$, while $\sum_{e \in \delta(S_1)} x_e = 1$ and $\sum_{e \in \delta(S_2)} x_e = 1$. Let $s_i := |\{v_1, v_2, v_3, v_4\} \cap S_i|$. It is straightforward to see that for $s_i \in \{0, 1, 3, 4\}$, both x' and x'' satisfy all blossom inequalities that are valid for x; similarly, the blossom inequalities are satisfied for $v_1, v_3 \in S_1$ and $v_2, v_4 \in \overline{S_1}$, as $\sum_{e \in \delta(S_1)} x'(e) > \sum_{e \in \delta(S_1)} x(e)$. Therefore, we only need to consider $v_1, v_2 \in S_1$ and $v_3, v_4 \in \overline{S_1}$, and $v_1, v_4 \in S_2$ and $v_2, v_3 \in \overline{S_2}$. Let $T_1 := S_1 \cap S_2$, $T_2 := S_1 \cap \overline{S_2}$, $T_3 := \overline{S_1} \cap \overline{S_2}$ $T_4 := \overline{S_1} \cap S_2$, and $E_{ij} := \{e = \{x, y\} \in E \mid x \in T_i, y \in T_j\}$. As $|T_1 \cup T_2|$, $|T_1 \cup T_4|$, $|T_2 \cup T_3|$, and $|T_3 \cup T_4|$ are all odd, we may assume without loss of generality that $|T_1|$ and $|T_3|$ are odd, and $|T_2|$ and $|T_4|$ are even.

Then we have

$$2 = \sum_{e\in\delta(S_1)} x_e + \sum_{e\in\delta(S_2)} x_e \tag{13}$$

$$= \sum_{e\in\delta(T_1)} x_e + \sum_{e\in\delta(T_3)} x_e + 2\sum_{e\in E_{24}} x_e \tag{14}$$

$$\geq \sum_{e\in\delta(T_1)} x_e + \sum_{e\in\delta(T_3)} x_e + 2x_{e_{24}} \tag{15}$$

$$> \sum_{e\in\delta(T_1)} x_e + \sum_{e\in\delta(T_3)} x_e. \tag{16}$$

This implies that $\min\{\sum_{e\in\delta(T_1)} x_e, \sum_{e\in\delta(T_3)} x_e\} < 1$, contradicting the assumption that x satisfies all cut inequalities. $\qquad\square$

The above lemma establishes the existence of optimal non-crossing matchings; when actually trying to find a non-crossing matching, it suffices to consider an additional term in the objective function of our LP that refers to the total length of the edges. This ensures that the total length of matching edges is minimized, avoiding crossings in the first place. For more details, see our experimental paper [12].

For the case of spanning trees, we can use a similar approach, based on a similar lemma.

Lemma 14 *For any set of vertices in the plane, there is a fractional spanning tree x of minimum stabbing number, such that the support graph of x is planar. Such a fractional spanning tree can be found in polynomial time.*

Proof. We proceed completely analogous to the proof of the previous lemma to deduce that if there was a fractional solution x of minimum total stabbing number among all solutions with optimal stabbing number, such that the support graph of x has two crossing edges, then we could shift an infinitesimal amount of weight from the crossing edges to two non-crossing ones, such that the stabbing number remains the same, but the total stabbing number decreases.

The technical steps of the argument are virtually identical: consider two crossing edges e_{13} and e_{24} as shown in Figure 12, and the alternative solutions x'_e and x''_e as defined in (11) and (12). For arguing that at least one of these solutions is feasible, we do not have to consider blossom constraints, but cut constraints of type (8).

Assume that for each of the two alternative solutions, a cut constraint is violated; by our assumption on ε, this means that there are two sets, $S_1 \subset P$ and $S_2 \subset P$, such that $\sum_{e\in\delta(S_1)} x'(e) < 1$ and $\sum_{e\in\delta(S_2)} x''(e) < 1$, while $\sum_{e\in\delta(S_1)} x_e = 1$ and $\sum_{e\in\delta(S_2)} x_e = 1$. Let $s_i := |\{v_1, v_2, v_3, v_4\} \cap S_i|$. It is straightforward to see that for $s_i \in \{0, 1, 3, 4\}$, both x' and x'' satisfy all cut inequalities that are valid for x; similarly, the cut inequalities are satisfied for $v_1, v_3 \in S_1$ and $v_2, v_4 \in \overline{S_1}$, as $\sum_{e\in\delta(S_1)} x'(e) > \sum_{e\in\delta(S_1)} x(e)$. Therefore, we only need to consider $v_1, v_2 \in S_1$ and $v_3, v_4 \in \overline{S_1}$, and $v_1, v_4 \in S_2$ and $v_2, v_3 \in \overline{S_2}$. Let $T_1 := S_1 \cap S_2$, $T_2 := S_1 \cap \overline{S_2}$, $T_3 := \overline{S_1} \cap \overline{S_2}$ $T_4 := \overline{S_1} \cap S_2$, and $E_{ij} := \{e = \{x, y\} \in E \mid x \in T_i, y \in T_j\}$. Using a sequence of estimates that is completely identical to (13)–(16), we get $2 > \sum_{e\in\delta(T_1)} x_e + \sum_{e\in\delta(T_3)} x_e$, again implying that $\min\{\sum_{e\in\delta(T_1)} x_e, \sum_{e\in\delta(T_3)} x_e\} < 1$, contradicting the assumption that x satisfies all blossom inequalities. $\qquad\square$

Theorem 15 *For any even set of vertices in the plane, there is a fractional perfect matching x of minimum stabbing number that has an edge of weight at least $1/5$. For any set of vertices in the plane, there is a fractional spanning tree x of minimum stabbing number that has an edge of weight more than $1/3$.*

Proof. For both problems, consider a fractional vertex with a planar support graph. To see the claim for matchings, note that there must be a vertex with degree at most five; as the total weight for each vertex is 1, the claim follows. To see the claim for spanning trees, note that the total edge weight is $n - 1$, and the number of edges is at most $3n - 6$, implying that the average weight is larger than $1/3$. □

Theorem 15 provides the basic ingredient for an iterated rounding algorithm: at each iteration, fix the weight of an edge of maximum fractional weight to one, and re-solve the linear program. In each iteration, the number of edges with fractional weight is reduced, so we get an overall polynomial-time algorithm for finding an integral solution.

Unfortunately, Jain's original proof only guarantees a constant-factor approximation for objective functions that arise as the (weighted) sum of the edge variables. However, the situation is different for our objective function which is a *maximum* over certain sums of edge variables, so an additional argument is needed for establishing a constant-factor guarantee. We are hopeful that this argument can be completed some time in the future [18]. As we demonstrate in our experimental paper [12], the practical performance seems to be even better than the theoretically possible guarantees of 5 and 3.

6   Notes and Conclusion

We have presented the first algorithmic paper on stabbing numbers, resolving the long-standing open question of complexity, and providing an approach that appears to be useful in theory and in practice. There are a number of interesting open questions.

Our proofs rely on a strong degeneracy of the vertex set, and it would be interesting to see a proof for vertices in general position.

We were not able to extend our $\mathcal{NP}$-hardness proof to the case of finding a triangulation of minimum general crossing number; another interesting issue is how to convert the $\mathcal{NP}$-hardness proof for triangulations of minimum crossing number into an $\mathcal{NP}$-hardness proof for minimum stabbing number.

Probably the most intriguing open question spawned by our work is whether the iterated rounding scheme suggested by the existence of a heavy edge in an optimal fractional solution to our linear programs (Lemmas 13 and 14) does indeed lead to a constant-factor approximation algorithm. Also, the use of the ellipsoid method (at least as a theoretical argument) is not "combinatorial," which always has to be considered a drawback.

Another interesting question is to decide the existence of structures of small constant stabbing number. As the hardness proof for deciding the existence of a matching of stabbing number 5 illustrates, this is still not an easy task. From some solvable special cases, we only note one; for a proof, see [12].

Theorem 16 *St-Tre$_2$(P)=2 and St-Mat(P)=2 can be decided in polynomial time.*

One may also ask for minimizing the *average* instead of the maximum stabbing number, and refer to the average over the whole continuum of lines intersecting a set of line segments, instead of just a combinatorial set of representatives. This, however, amounts to solving problems of minimum length, with all implications to hardness and approximation; again, see [12] for details.

Theorem 17 *A set of line segments has minimum average (axis-parallel, resp.) stabbing number with respect to uniform distribution of lines, if and only if the overall Euclidean (Manhattan, resp.) length of all line segments is minimum.*

We remark that a linear program for minimizing the average stabbing number can be written with a sum in the objective function (instead of a maximum as we had to model it), allowing us to directly apply our iterated rounding technique and obtaining the desired approximation factors of 3 and 5, respectively; see [12].

Acknowledgments

References

[1] P.K. Agarwal. Ray shooting and other applications of spanning trees with low stabbing number. *SIAM J. Computing*, 21(3):540–570, 1992.

[2] P.K. Agarwal, B. Aronov, and S. Suri. Stabbing triangulations by lines in 3D. In *Proc. 11th ACM Sympos. Computational Geometry*, pages 267–276, 1995.

[3] E. M. Arkin, M. A. Bender, E. Demaine, S. P. Fekete, J. S. B. Mitchell, and S. Sethia. Optimal covering tours with turn costs. *SIAM Journal on Computing*, 35:531–566, 2005.

[4] B. Aronov, H. Brönnimann, A.Y. Chang, and Y.-J. Chiang. Cost-driven octree construction schemes: an experimental study. *Comput. Geom. Theory Appl.*, 31:127–148, 2005.

[5] B. Aronov and S. Fortune. Approximating minimum-weight triangulations in three dimensions. *Discrete Comput. Geom.*, 21(4):527–549, 1999.

[6] Bernard Chazelle and Emo Welzl. Quasi-optimal range searching in space of finite vc-dimension. *DISCG*, 4:467–489, 1989.

[7] M. de Berg and M. van Kreveld. Rectilinear decompositions with low stabbing number. *Inform. Process. Lett.*, 52(4):215–221, 1994.

[8] E.D. Demaine, J.S.B. Mitchell, and J. O'Rourke. The open problems project. http://cs.smith.edu/~orourke/TOPP/Welcome.html, 2003.

[9] J. Edmonds. Maximum matching and a polyhedron with 0,1-vertices. *J. Res. Nat. Bur. Standards*, 69B:125–130, 1965.

[10] S. P. Fekete. On simple polygonalizations with optimal area. *Discrete Comput. Geom.*, 23:73–110, 2000.

[11] S. P. Fekete, M. E. Lübbecke, and H. Meijer. Minimizing the stabbing number of matchings, spanning trees, and triangulations. In *Proc. 15th ACM-SIAM Sympos. Discrete Algorithms*, pages 430–439, 2004.

[12] S. P. Fekete, M. E. Lübbecke, and H. Meijer. Computing structures of minimum stabbing number. Technical report, 2008.

[13] M.R. Garey and D.S. Johnson. *Computers and Intractability—A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, San Francisco, 1979.

[14] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, Berlin, 1988.

[15] M. Held, J.T. Klosowski, and J.S.B. Mitchell. Evaluation of collision detection methods for virtual reality fly-throughs. In *Proc. 7th Canadian Conf. Computational Geometry*, pages 205–210, 1995.

[16] J. Hershberger and S. Suri. A pedestrian approach to ray shooting: Shoot a ray, take a walk. *J. Algorithms*, 18:403–431, 1995.

[17] K. Jain. A factor 2 approximation algorithm for the generalized Steiner network problem. *Combinatorica*, 21(1):39–60, 2001.

[18] K. Jain. Personal communication, 2003.

[19] T.L. Magnanti and L.A. Wolsey. Optimal trees. In M.O. Ball, T.L. Magnanti, C.L. Monma, and G.L. Nemhauser, editors, *Network Models*, volume 7 of *Handbooks in Operations Research and Management Science*, chapter 9, pages 503–616. North-Holland, 1995.

[20] J. Matoušek. Spanning trees with low crossing number. *Inform. Theor. Appl.*, 25:102–123, 1991.

[21] J.S.B. Mitchell and J. O'Rourke. Computational geometry column 42. *Int. J. Comput. Geom. Appl.*, 11(5):573–582, 2001.

[22] M. Padberg and M.R. Rao. Odd minimum cut-sets and b-matchings. *Mathematics of Operations Research*, 7:67–80, 1982.

[23] A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*. Springer, Berlin, 2003.

[24] J.R. Shewchuk. Stabbing Delaunay tetrahedralizations. *Discrete Comput. Geom.*, 32(3):339–343, 2004.

[25] C. Tóth. Orthogonal subdivisions with low stabbing numbers. In *Proc. 9th International Workshop on Algorithms and Data Structures (WADS 2005)*, volume 3608 of *Springer LNCS*, pages 256–268, 2005.

[26] E. Welzl. On spanning trees with low crossing numbers. In B. Monien and Th. Ottmann, editors, *Data Structures and Efficient Algorithms*, volume 594 of *Springer LNCS*, Berlin, 1992.