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. We investigate problems of finding perfect matchings, spanning trees, or triangulations of minimum stabbing number for a given set of points. The complexity of these problems has been a long-standing open problem; in fact, it is one of the original 30 outstanding open problems in computational geometry on the list by Demaine, Mitchell, and O'Rourke.

We show that minimum stabbing problems are NP-complete. We also show that an iterated rounding technique is applicable for matchings and spanning trees of minimum stabbing number by showing that there is a polynomially solvable LP-relaxation that has fractional solutions with at least one heavy edge. This suggests constant-factor approximations. Our approach uses polyhedral methods that are related to another open problem (from a combinatorial optimization list), in combination with geometric properties. We also demonstrate that the resulting techniques are practical for actually solving problems with up to several hundred points optimally or near-optimally.

Classification: F.2.2 Nonnumerical Algorithms and Problems

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

1 Introduction

Objective Functions. Typical problems in combinatorial optimization, algorithmic graph theory, or computational geometry deal with minimizing the length of a desired structure: Given a set of points, find a set of line segments of small total length, such that a certain structural condition is maintained. Among the most popular such structures are spanning trees, perfect matchings, or (in a planar geometric setting) triangulations. However, some geometric scenarios motivate other objective functions; one such alternative for measuring the quality of a structure is the total turn cost between adjacent line segments; e.g., see [3].

When dealing with structural or algorithmic properties, one can be more interested in yet another objective function called the *stabbing number*. In order to unify definitions for different structures and to allow for a consistent notation throughout this paper, we describe this as a property of a set of line segments: For a given set of line segments, the *stabbing number* is the maximum number of segments that are encountered (in their interior or at an endpoint) by any infinite line; if we consider only axis-parallel lines, we get the *axis-parallel stabbing number*. When focusing on the number of objects defined by the line segments, we may consider the closely related *crossing number*, which arises from considering the number of connected components in the intersection with the set of line segments that we have to cross along a line. In the absence of connected components of collinear segments (which is the case for matchings), the crossing number is equal to 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.

Related Work. Stabbing problems have been considered for a number of years. The complexity of many algorithms in computational geometry is directly dependent on the complexity of ray-shooting, which depends directly on the stabbing number. Agarwal [1] describes several applications of spanning trees with low stabbing number, among them *ray-shooting* and *implicit point lo-*

[*] Department of Mathematical Optimization, Braunschweig University of Technology, Pockelsstraße 14, D-38106 Braunschweig, Germany. Email: s.fekete@tu-bs.de.

[†] Institut für Mathematik, Sekr. MA 6-1, Technische Universität Berlin, Straße des 17. Juni 136, D-10623 Berlin, Germany. Email: m.luebbecke@math.tu-berlin.de. Visits to Kingston and Stony Brook in May 2002 were supported by a DFG travel grant.

[‡] Dept. of Computing and Information Science, Queen's University, Kingston, Ontario, K7L 3N6, Canada. Email: henk@cs.queensu.ca. Partially supported by NSERC while visiting Braunschweig during fall 2002.

cations queries (which by themselves have applications in polygon containment, implicit hidden surface removal, polygon placement, etc.). The theoretically best performing data structure for ray tracing in two dimensions is based on a triangulation of the scene; see Hershberger and Suri [11]. Agarwal, Aronov, and Suri [2] investigate the stabbing number of triangulations in three dimensions, where the stabbed objects are simplices. Held, Klosowski, and Mitchell [10] investigate collision detection in a virtual reality environment; again, we have a dependency on the stabbing number.

Extremal properties of crossing numbers were considered by Welzl [25] and by Matoušek[17], who showed that any planar set of n points has a spanning tree with a crossing number of $O(\sqrt{n})$, and there are examples requiring a crossing number of $\Omega(\sqrt{n})$. Another variant is studied by de Berg and van Kreveld [5]: 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. Shewchuk [23] 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.

Despite of this interest by a fair number of noteworthy researchers, there have been no results or conjectures whatsoever on the *complexity* of stabbing problems. In fact, resolving the complexity of stabbing problems has been one of the original 30 outstanding open problems of computational geometry on the list that was first presented by Mitchell and O'Rourke in [19], based on many years of preceding discussions and informal surveys. An up-to-date and expanded list is maintained by Demaine, Mitchell, and O'Rourke and can be found on the internet [6]. In addition to the open complexity status, the reader should take note that stabbing problems have defied all attempts for obtaining good combinatorial lower bounds, and nothing is known about approximation, other than the factor $O(\sqrt{n})$ that can be deduced from Welzl's work.

Our Work. In this paper, we present the first general algorithmic study of stabbing problems. We resolve the open problem of complexity for various structures; our technique is quite general, and it seems clear that it can be extended to other versions. We also describe a general technique based on Linear Programming (LP) that yields good lower bounds, and is closely related to an open problem from another prominent list, this time from the combinatorial optimization community [14, 15]. As a consequence, we are able to deduce an extension of the iterated rounding technique developed by Jain for generalized Steiner network problems [12]; making additional use of the geometric structure of the problem, we get the basis for constant-factor approximations. Finally, we demonstrate by a computational study on various benchmark sets that our LP-based approach is also practically useful, both for solving problems approximately or optimally, by considering integer programming (IP).

Our results in detail:

- We prove that deciding whether a point set has a perfect matching of axis-parallel stabbing number 5 is an NP-complete problem; we also extend this result to general stabbing number.
- We give an NP-completeness proof for finding a spanning tree of axis-parallel or general stabbing number, and hint at a hardness proof for axis-parallel or general crossing number.
- We prove that finding a triangulation of minimum axis-parallel crossing number is NP-hard.
- We describe an LP-based class of lower bounds that can be evaluated in polynomial time. From a theoretical point of view, we use the ellipsoid method; the existence of a strongly polynomial algorithm for a closely related class of problems is subject to an open problem from the list [14, 15].
- 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 allows applying an iterated rounding technique, similar to the one developed by Jain for generalized Steiner network problems; this should imply constant-factor approximations.
- We describe the results of a computational study. Using a diverse set of benchmark instances (based on TSPLIB, Solomon's vehicle routing problems, and two different types of random instances) we are able to compute optimal and near-optimal solutions for instances up to several hundred points. This demonstrates 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). Results indicate far better

approximation quality than the theorically possible factors of 5 or 3, respectively.

It should be noted that our positive (LP-based) results do *not* make any assumptions on the structure of the point set: They can be used for point 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. On the other hand, the point sets constructed in our hardness proofs make use of collinear points.

The rest of this paper is organized as follows. After some basic definitions and notation in Section 2, we give sketches of our various hardness proofs in Section 3, with details omitted from this extended abstract. 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; the resulting algorithms appear to be constant-factor approximations. Section 6 presents a detailed computational study on perfect matchings of low stabbing number. Final concluding thoughts and miscellaneous results and problems are presented in Section 7.

2 Preliminaries

In the following, we consider a planar point set P of cardinality n. When dealing with matchings, we assume that n is even, if necessary by omitting one of the points.

Given a set of line segments L, 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 the rest of this paper, the set L will arise as a matching, spanning tree, or triangulation of a planar point set P, and our objective is to find such a structure of minimum stabbing number. We denote by $\mathit{St\text{-}Mat}_{all}(P)$ the minimum stabbing number among all matchings of P, and by $\mathit{St\text{-}Mat}_2(P)$ the minimum axis-parallel stabbing number of all matching of P. Similarly, we denote by $\mathit{St\text{-}Tre}_{all}(P)$ the minimum stabbing number of all spanning trees of P, by $\mathit{St\text{-}Tre}_2(P)$ the minimum axis-parallel stabbing number of all spanning trees of P; by $\mathit{St}\text{-}\Delta_{all}(P)$ the minimum stabbing number of all triangulations of P, and by $\mathit{St}\text{-}\Delta_2(P)$ the minimum axis-parallel stabbing number of all triangulations of P.

If we are not interested in the number of line segments in L encountered by a line ℓ, but in the number of connected components of $L \cap \ell$, we get the *crossing number*. For matchings, trees, and triangulations, we use the analogous abbreviations $\mathit{Cr\text{-}Mat}_{all}(P)$, $\mathit{Cr\text{-}Mat}_2(P)$, $\mathit{Cr\text{-}Tre}_{all}(P)$, $\mathit{Cr\text{-}Tre}_2(P)$, $\mathit{Cr}\text{-}\Delta_{all}(P)$, $\mathit{Cr}\text{-}\Delta_2(P)$. Note that stabbing and crossing number or depth are the same for matchings.

3 Complexity

In this section we give proofs and proof sketches that virtually all variants of minimum stabbing problems are NP-hard. Our technique is rather general and should be applicable to other variants as well. In this extended abstract, we give only proof sketches; full details can be found in the full version of the paper.

3.1 Matchings

THEOREM 3.1. *Deciding whether* $\mathit{St\text{-}Mat}_2(P) \leq 5$ *is a strongly NP-complete problem.*

Proof. We prove the theorem by using a reduction from 3SAT [9]. 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. We construct a set of points P that has a matching M of stabbing number 5 if and only if the Boolean expression can be satisfied. For the overall layout see Figure 1. Note the collinear sets of points that function as "barriers": As they already require a stabbing number of 5, they must not be crossed by additional line segments, thus imposing a clear combinatorial structure that is exploited in the proof.

Each variable gadget allows two feasible matchings; the particular choice represents a truth assignment for a particular variable, which in turn imposes requirements on how the literal gadgets in the respective columns may be chosen. Clauses are represented by three literal gadgets in the same row; the overall construction implies that the stabbing number of the row of a clause is at most five, iff at least one literal gadget in this row contributes only one stabbed line segment, meaning that at least one literal satisfies the clause. □

COROLLARY 3.1. *There is no* α*-approximation algorithm for* $\mathit{St\text{-}Mat}_2(P)$ *with* $\alpha < 6/5$.

COROLLARY 3.2. *Computing* $\mathit{St\text{-}Mat}_{all}(P)$ *is a weakly NP-hard problem.*

Proof. We apply a perturbation technique, similar to the one in [8]. Use the same construction as for the hardness proof for the axis-parallel case. Consider the grid formed by the coordinates of the resulting point set. This grid is modified such that the interpoint distances between the points of the same gadget are $\Theta(\varepsilon^{n^2})$. Furthermore, the rest of the grid is perturbed by powers of ε, such that only axis-parallel lines can stab more than two gadgets. Now it is easy to see that only axis-parallel lines are critical. □

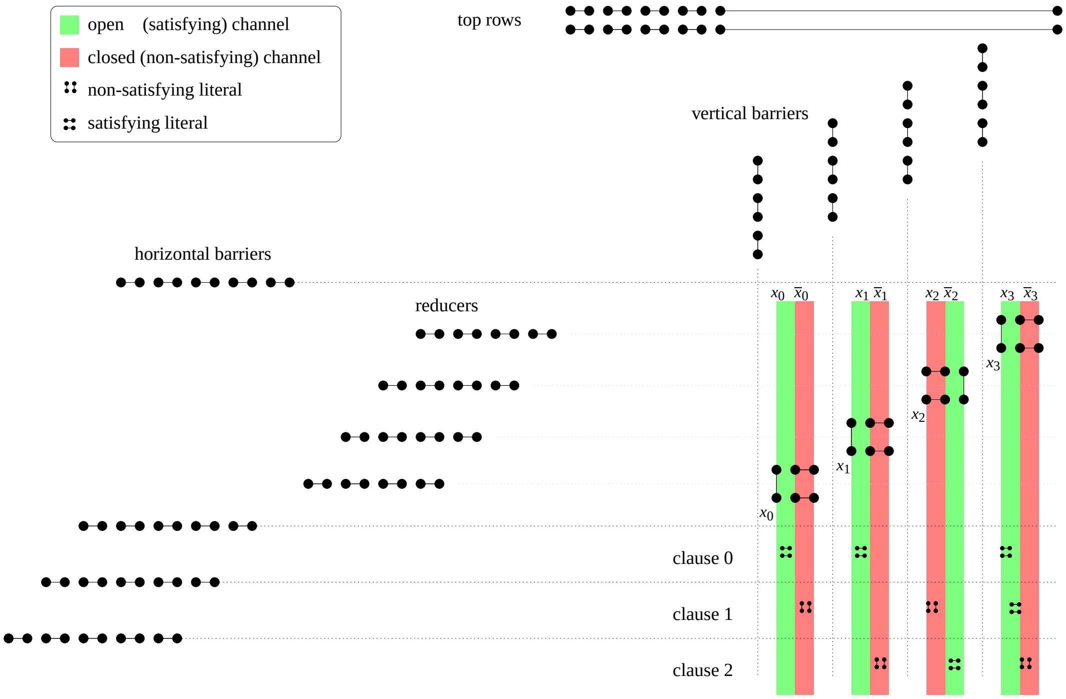

Figure 1: Overall layout for $\mathit{St\text{-}Mat}_2(P)$.

3.2 Spanning Trees. The basic construction for showing hardness of finding a spanning tree of minimum stabbing number is very similar to the one for matchings. As before, we use barriers to restrict possible connections: We make use of the arrangement shown (in horizontal orientation) in Figure 2.

LEMMA 3.3. *Let S be the arrangement of $3k$ points shown in Figure 2, and let $P \supseteq S$ have no other points in the horizontal strip indicated by shading. If P has a spanning tree T with stabbing number $c+1$, then no edge of T crosses the shaded region.*

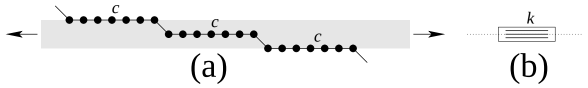

Figure 2: A barrier gadget: (a) In a spanning tree of stabbing number $c+1$, no line segment may cross the shaded region. (b) Symbol for the barrier gadget; the dotted line indicates the blocked strip.

The variable gadgets look as in Figure 3. Note the use of barrier gadgets, and the remaining numbers of segments that may cross the induced dotted lines.

LEMMA 3.4. *Let S be the arrangement of points shown in Figure 3, with barrier gadgets placed and sized as indicated, and let $P \supseteq S$. Then any spanning tree of P that has stabbing number at most $k+1$ uses either the two "true" or the two "false" edges.*

The overall layout of the construction is shown in Figure 4. For proving hardness of finding a spanning tree of minimum crossing number we use the following barrier gadget. For a proof note that each elementary 2×2 cycle of the gadget must have at least one edge not present in a tree; then the claim follows by pigeonhole principle. Finally, the NP-hardness proof for all directions follows again by making only the axis-parallel directions to be critical.

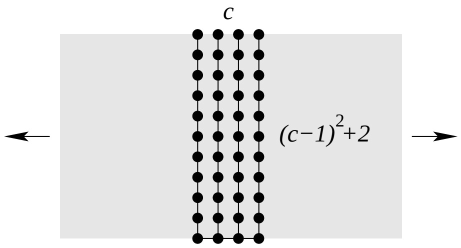

Figure 5: A barrier gadget for showing hardness of minimizing the crossing number of a spanning tree.

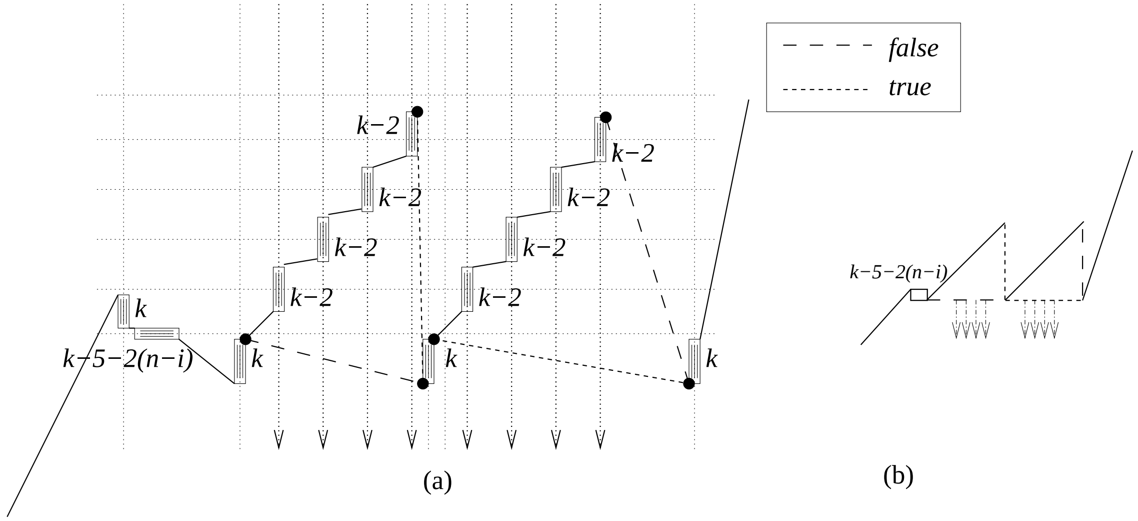

Figure 3: A gadget for variable x_i: (a) In a spanning tree of stabbing number $k+1$, either the “true” or the “false” setting is chosen. (b) Symbol for the variable gadget.

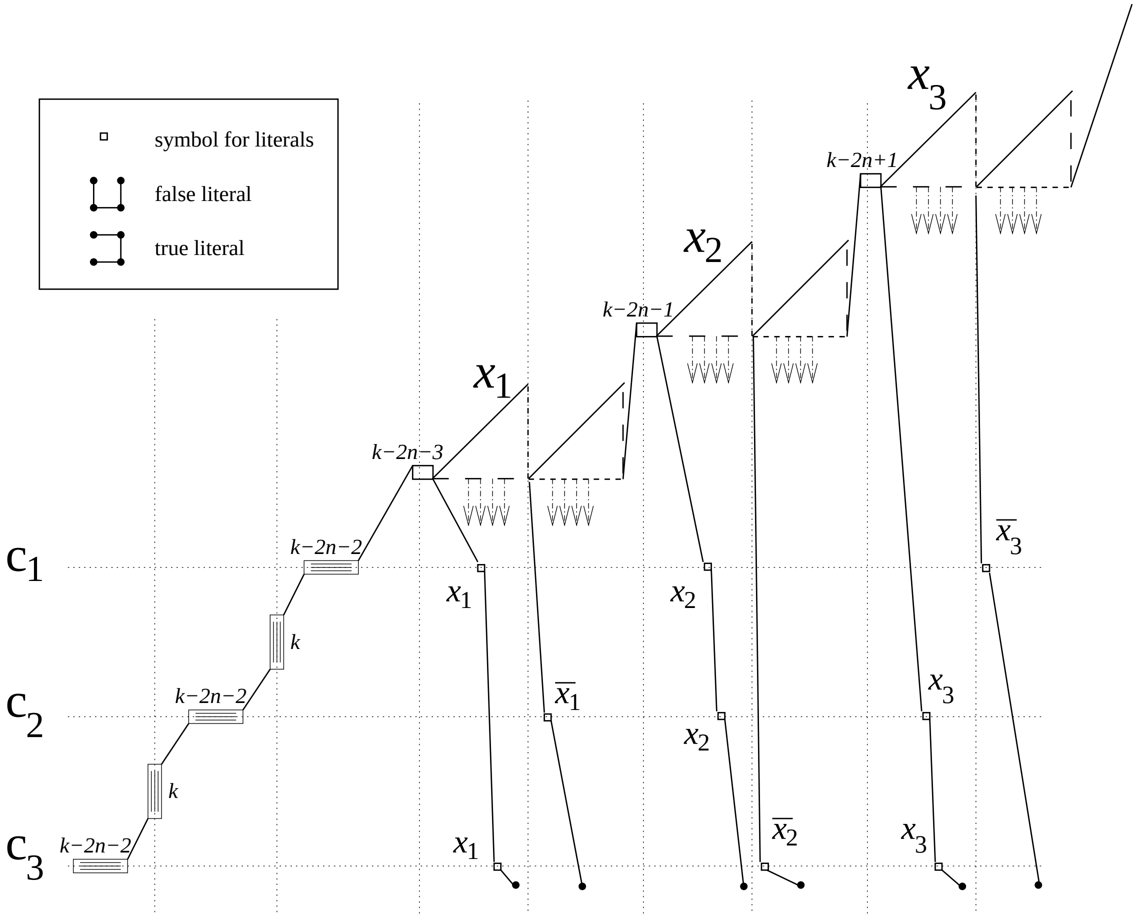

Figure 4: The overall layout for the hardness proof for spanning trees. n is the total number of variables, k a sufficiently large number; in question is the existence of a spanning tree with stabbing number $k+1$. Shown is the representation of the 3SAT instance $(x_1 \vee x_2 \vee \overline{x_3}) \wedge (\overline{x_1} \vee x_2 \vee x_3) \wedge (x_1 \vee \overline{x_2} \vee x_3)$, for $n=3$.

LEMMA 3.5. *Let S be the $k \times (c-1)^2 + c$ arrangement of points shown in Figure 5, and let $P \supseteq S$. If P has a spanning tree T with crossing number c, then no other edge of T crosses the shaded region.*

Summarizing, we state:

THEOREM 3.2. *It is NP-hard to determine St-Tre$_2(P)$, St-Tre$_{all}(P)$, Cr-Tre $_2(P)$, or Cr-Tre $_{all}(P)$.*

3.3 Triangulations. We use the following terminology. A *horizontal line* is a set of points that are on a horizontal line. A *vertical line* is a set of points that are on a vertical line. 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 3.6. *Consider a row consisting of two horizontal lines in P, having a and b points, respectively. The horizontal stabber of this row encounters at least $a+b-2$ triangles in any triangulation of P.*

A detailed proof can be found in the full paper. The lemma analogously holds for two vertical lines that form a column. When a row consists of two horizontal lines that have *Cr-*Δ $_2(P)+2$ points altogether, we call it *full* or fully triangulated. In a triangulation that achieves the lower bound of the lemma this row has the property that all edges along the horizontal lines have to be present. The same applies to full columns, and this is the way how we will use the lemma.

THEOREM 3.3. *Finding Cr-*Δ $_2(P)$ *is NP-hard.*

The proof is based on another reduction of 3SAT. The mechanics of gadgets is similar to our previous two reductions. Details can be found in the full version of the paper. See Figure 6 for a schematic layout of a representing point set P for the 3SAT instance $B(x_0, x_1, x_2) = (x_0 \vee x_1 \vee \overline{x_2}) \wedge (x_0 \vee \overline{x_1} \vee x_2) \wedge (\overline{x_0} \vee \overline{x_1} \vee \overline{x_2})$.

4 Linear Programs for Stabbing

4.1 Linear Programs and *St-Mat*(P)**.** Thinking of P as the vertex set V of a straight-line embedded complete graph $G = (V, E)$, a handsome representation of a perfect matching M is by its edge incidence vector $x \in \{0,1\}^E$, where $x_{ij} = 1$ if $ij \in M$ and $x_{ij} = 0$ otherwise. For $S \subseteq V$ denote by $\delta(S) = \{ij \in E \mid i \in S, j \notin S\}$. The following linear inequalities are necessarily satisfied for a perfect matching given by x.

$$\sum_{ij \in \delta(i)} x_{ij} = 1 \quad \forall i \in V \tag{4.1}$$

$$\sum_{ij \in \delta(S)} x_{ij} \geq 1 \quad \forall S \subseteq V,\ |S| \text{ odd} \tag{4.2}$$

$$x \geq 0 \tag{4.3}$$

In a seminal paper Edmonds [7] showed that this polyhedral description is also sufficient in the sense that the extreme points of the polytope Π defined by (4.1)–(4.3) are exactly the incidence vectors of perfect matchings in G. Despite the fact that there are exponentially many so-called *blossom inequalities* (4.2) one can solve a linear program over Π in strongly polynomial time [4].

Now we wish to minimize the stabbing number k and add to Π

$$\sum_{ij : ij \cap \ell(d) \neq \emptyset} x_{ij} \leq k \qquad \forall \text{ stabbing line } \ell(d) \text{ in direction } d. \tag{4.4}$$

In principle, there are infinitely many constraints of this type, even for one direction. Note, however, that when sweeping a stabbing line in direction d the stabbing number changes only at a vertex. Therefore, we only need to check a linear number of lines in each direction. For the same reason, "all" directions reduce to the $O(n^2)$ combinatorial directions determined by all pairs of vertices of G. When requiring integrality of x and minimizing k in the resulting integer program yields exactly *St-Mat*(P). When integrality is relaxed to (4.3), this linear programming relaxation gives a lower bound on *St-Mat*(P). The solution x will in general be fractional, and we speak of *fractional stabbing number* in this context.

The resulting LP can be solved in weakly polynomial time by means of the ellipsoid method [18]: Separating violated blossom inequalities is possible in polynomial time [20], and there are only polynomially many stabbing constraints. It should be noted that this is closely related to another well-considered open problem: [14, 15] asks for a strongly polynomial algorithm for finding an optimal matching in the presence of a single general "budget" constraint, which is a general version of stabbing constraints. This illustrates that giving a strongly polynomial algorithm for our class of LPs may not be an easy task, even though it is the intersection of two well-behaved polyhedra. This difficulty has been known for other classes of intersecting polyhedra [22].

4.2 Linear Programs and *St-Tre*(P)**.** There are several polynomial-size formulations for spanning trees [16]. However, even though exponential in size, the following cut-based LP formulation turns out to have some particularly useful properties:

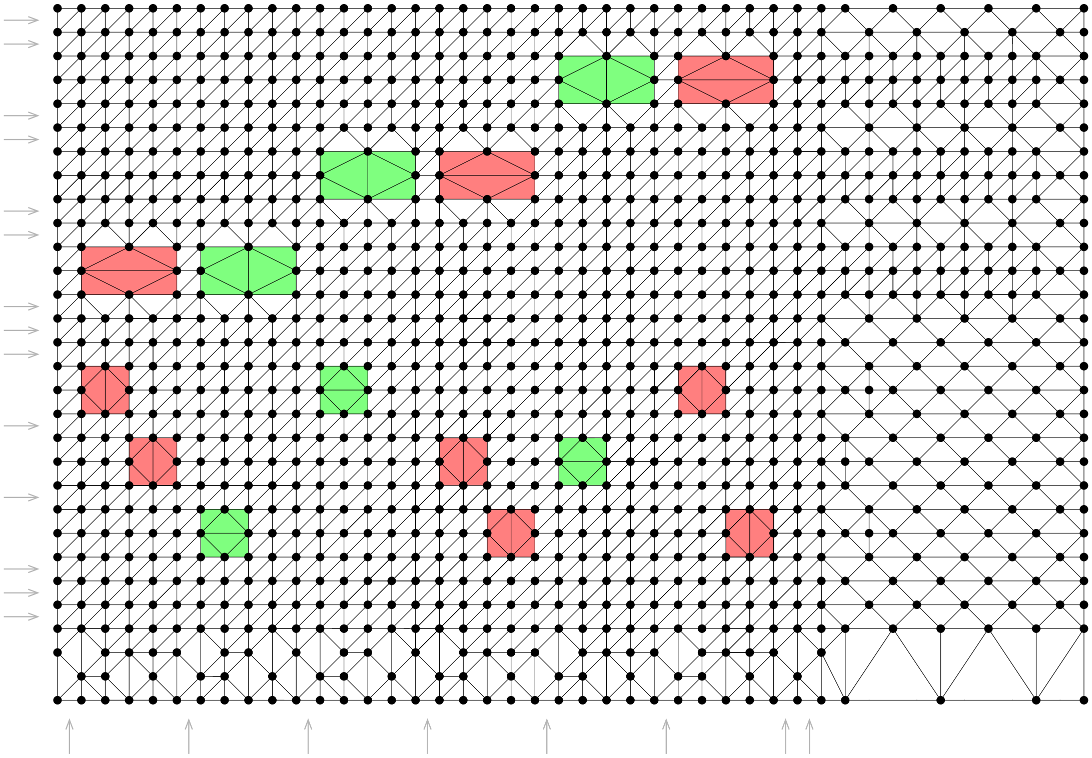

Figure 6: Overall layout for *Cr*-$\Delta_2(P)$. Clauses are $(x_0 \vee x_1 \vee \overline{x_2})$, $(x_0 \vee \overline{x_1} \vee x_2)$, and $(\overline{x_0} \vee \overline{x_1} \vee \overline{x_2})$, with $x_0 =$ *false* and $x_1 = x_2 =$ *true*. Arrows indicate full rows and columns, light or dark shading indicates true or false variables and literals.

$$\sum_{ij\in E} x_{ij} = n-1 \tag{4.5}$$

$$\sum_{ij\in\delta(S)} x_{ij} \geq 1 \quad \forall S \subseteq V, \tag{4.6}$$

$$\sum_{ij\in E(S)} x_{ij} \leq |S|-1 \quad \forall S \subseteq V, \tag{4.7}$$

$$x \geq 0 \tag{4.8}$$

with the additional stabbing constraints

$$\sum_{ij:ij\cap\ell(d)\neq\emptyset} x_{ij} \leq k \qquad \forall \text{ stabbing line } \ell(d) \text{ in direction } d. \tag{4.9}$$

Again, solving this LP when minimizing k can be achieved in (weakly) polynomial time.

5 Iterated Rounding

Key ingredients of the linear programs described in the previous section are constraints having a *cut structure*: We require that for any set from a given family of subsets of vertices, there is a guaranteed lower bound on the size of a cut. Integer programs of this type have been called *generalized Steiner network problems*. It is known that these problems do not tend to have nicely structured fractional vertices. This does not get better in the presence of stabbing constraints. In the full paper we provide examples with edges of rather small fractional weight. This prohibits a straightforward approximation by simply rounding up a fractional solution.

A very elegant general technique that overcomes these difficulties and achieves a 2-approximation algorithm for generalized Steiner networks problems was presented by Jain in [12]: Based on a polyhedral argument, he established that any fractional solution of a generalized Steiner network LP must have an edge of weight at least 1/2. From this, he derived an approximation algorithm by iteratively rounding up the weight of the heaviest edges, and re-solving the LP with these fixed edge weights.

It is natural to consider such an approach for our LPs for deriving approximation methods for stabbing problems, in particular for matchings and spanning trees, as the stabbing constraints also have cut structure. However, Jain's crucial lemma guaranteeing a heavy edge in a fractional solution does no longer apply in the presence of stabbing constraints: Figure 7 shows a matching instance for which the optimal fractional axis-parallel stabbing number is achieved by a unique solution with maximum edge weight 1/3.

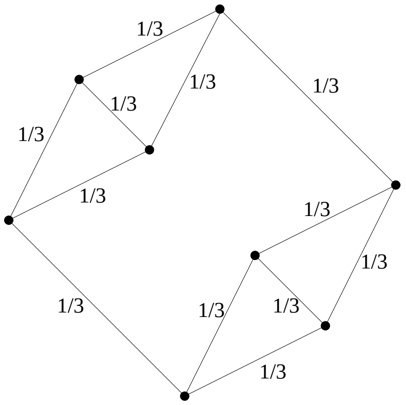

Figure 7: An optimal fractional solution with maximum edge weight 1/3.

However, using additional geometric properties, we can still establish lower bounds. Our proofs are based on the following lemmas.

LEMMA 5.1. *For any even set of points 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.*

LEMMA 5.2. *For any set of points 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.*

Both proofs use an uncrossing 2-exchange argument. Note that the proof for Lemma 5.1 requires some extra care because of the blossom inequalities. A detailed proof can be found in the full paper. The proof of Lemma 5.2 is almost completely analogous.

THEOREM 5.1. *For any even set of points 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 points 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. □

This sets the stage for an iterated rounding procedure: At each iteration, fix the weight of an edge of maximum fractional weight to 1, and resolve the linear program. In each iteration, the number of edges with fractional weight is reduced, so we get an overall polynomial method for finding an integral solution.

It should be noted that even though the key ingredient for an iterated rounding procedure is provided by Theorem 5.1, there is still one element missing for establishing that these polynomial algorithms are indeed constant-factor approximation algorithms: As the objective function is a maximum, not a sum, we still need an extra argument to assure that Jain's overall approach does indeed work. We are hopeful that this argument can be completed [13]. As we show in the following Section 6, the practical performance seems to be even better than the theoretically possible guarantees of 5 and 3.

6 Computational Results

Test suite. We compiled a test suite of various instances on which we evaluated our LP/IP approaches and the iterated rounding technique for $St\text{-}Mat_2(P)$. The suite includes ten instances with up to 442 points from the TSPLIB[21]; the C-class ("clustered") of Solomon's instances of the vehicle routing problem [24] with 100 points each; 25 regular grids with 20 to 360 points, based on grids of size 5×5 up to 20×20 in which 20% randomly chosen points are removed; and a set of instances with up to 100 random points in the plane. Tables 1 and 2 display our preliminary results on a Pentium III 700MHz PC with 1GB main memory running Linux. LPs and IPs are solved by CPLEX8.0, CPU times are listed in seconds. Table entries bearing the † sign indicate an exceeding of our CPU time limit of four hours for solving the IP, i.e., computing the exact optimum.

Some brief observations. In LP solutions variables may assume rather arbitrary fractional and small values; this is also true when blossom inequalities are added. The colinearity of points in the grid instances enables us to reduce the number of stabbing constraints, resulting in significantly reduced computation times. The clustering of points in the vehicle routing instances obviously facilitate the LP/IP solution process, as was to be expected. However, this observation is interesting in practice where the data is usually well structured as opposed to randomly distributed.

Instance	LP opt	LP CPU	IP opt	IP CPU	iter. rounding
ulysses22	1.992308	0.01	2	0.05	2
berlin52	2.815158	0.52	4	8.86	5
lin105	5.500000	1.21	infeas.	†	7
bier127	4.330856	5.64	infeas.	†	5
u159	15.000000	1.76	infeas.	†	15
ts225	13.750000	5.37	infeas.	†	16
tsp225	11.500000	62.06	infeas.	†	12
a280	10.500000	55.12	$\leq$12	†	13
lin318	8.113143	131.54	$\leq$12	†	11
pcb442	16.500000	270.03	$\leq$19	†	18
c101	7.000000	0.10	7	1.75	8
c102	7.000000	0.10	7	1.75	8
c103	7.000000	0.10	7	1.75	8
c104	7.000000	0.10	7	1.75	8
c105	7.000000	0.10	7	1.75	8
c106	7.000000	0.10	7	1.75	8
c107	7.000000	0.10	7	1.75	8
c108	7.000000	0.10	7	1.75	8
c201	6.000000	0.50	6	2.00	7
c202	6.000000	0.45	6	2.00	7
c203	6.000000	0.46	6	1.95	7
c204	6.000000	0.46	6	1.95	7
c205	6.000000	0.46	6	1.95	7
c206	6.000000	0.46	6	1.92	7
c207	6.000000	0.46	6	1.95	7
c208	2.953237	0.58		†	5

Table 1: Comparison of bounds for $\mathit{St\text{-}Mat}_2(P)$ and iterated rounding: TSPLIB and clustered instances.

Instance	LP opt	LP CPU	IP opt	IP CPU	iter. rounding
grid5a	2.500000	0.00	3	0.01	3
grid5b	2.750000	0.00	3	0.02	3
grid5c	2.750000	0.01	3	0.02	4
grid5d	2.000000	0.00	3	0.03	3
grid5e	2.500000	0.00	3	0.02	3
grid8a	5.003205	0.04	6	0.17	6
grid8b	5.125000	0.05	6	0.21	6
grid8c	5.000000	0.04	5	0.12	6
grid8d	5.428571	0.04	6	0.19	7
grid8e	5.403226	0.04	6	0.29	6
grid10a	4.250000	0.12	5	0.68	6
grid10b	4.250000	0.10	5	0.40	6
grid10c	5.250000	0.11	6	245.28	6
grid10d	4.500000	0.07	5	13.58	6
grid10e	5.000000	0.03	5	1.36	6
grid15a	6.000000	0.73	6	8.40	7
grid15b	7.500000	0.47	8	9.90	8
grid15c	6.000000	1.00	6	3.27	6
grid15d	6.500000	0.24	7	†	7
grid15e	6.750000	0.74	7	3.91	7
grid20a	9.166667	15.38	$\leq$11	†	12
grid20b	9.250000	20.99	$\leq$11	†	11
grid20c	9.500000	8.70	$\leq$11	†	11
grid20d	9.500000	26.69	$\leq$11	†	11
grid20e	10.000000	20.43	$\leq$11	†	12
rand10a	1.750000	0.00	2	0.01	2
rand10b	1.833333	0.00	2	0.00	2
rand10c	1.750000	0.00	2	0.01	2
rand10d	1.700000	0.00	2	0.01	2
rand10e	1.812500	0.00	2	0.01	2
rand50a	2.594823	0.25	3	19.83	5
rand50b	2.628112	0.23	3	1.91	4
rand50c	2.668918	0.23	4	30.77	4
rand50d	2.661581	0.22	4	15.99	4
rand50e	2.789609	0.33	4	25.54	5
rand100a	3.376247	5.57		†	6

Table 2: Comparison of the LP/IP bounds for $\mathit{St\text{-}Mat}_2(P)$ and iterated rounding: grid and random instances.

The stabbing number obtained from iterated rounding is often very close to the LP lower bound, and in our experiments it is never off the optimal value by more than a factor of 2; typically, it is much better, and mostly within about 20% of the optimum. We observe that the "bad moves" are made only in the final iterations of the iterated rounding. For instance, about 100 iterations are needed for lin318 (with 318 points), the LP optimum is at 8.113, and the LP value exceeds 9.000 only in the last ten iterations, where a value of 11 is reached. We also experimented with a "one good shot at once" approach that is based on the fact that each fractional vertex is the convex combination of perfect matchings, by finding a maximum weight perfect matching in the support graph of the LP solution. This usually gives even better feasible solutions than for iterated rounding (with one exception in our test suite of problems). This technique certainly deserves further evaluation both from a computational and from a theoretical point of view, and a discussion will be included in the full paper.

The stabbing constraints seem to completely destroy the polyhedral structure of the matching polytope. Half of the TSP instances are infeasible (because of an odd number of points), and this is not detected by the CPLEX IP solver within 4 hours. Iterated rounding terminates in this case with a non-perfect matching with one point unmatched.

Even though our prototypes are not yet ready to solve even larger instances, they may serve as a good starting point for the development of an industrial strength code. We plan to include more computational results, in particular for $\mathit{St\text{-}Mat}_{all}(P)$ and $\mathit{St\text{-}Tre}_2(P)$, in the full paper. We also plan tests with IP formulations for triangulations.

7 Notes and Conclusion

We have shown that various versions of stabbing problems are NP-hard, and demonstrated how an IP/LP-based approach may be useful for solving and approximating instances. We expect a number of extensions of this work and hope to include more details and results in a forthcoming full journal version. Here we only mention a number of other aspects that are also possible.

Clearly the most interesting open problem is a proof that our iterated rounding technique is indeed a constant-factor approximation algorithm, with a performance guarantee of 5 for matchings, and 3 for spanning trees.

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:

THEOREM 7.1. *St-Tre*$_2(P)$*=2 and St-Mat*$_{all}(P)$*=2 can be decided in polynomial time.*

One may also ask for the *average* instead of the maximum stabbing number, and refer to the average over all possible 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.

THEOREM 7.2. *A set of line segments has minimum average (axis-parallel, resp.) stabbing number, iff the overall Euclidean (Manhattan, resp.) length of all line segments is minimum.*

Finally, the closely related question of investigating maximum stabbing-independent subsets, i.e., subsets of lines with stabbing number 1, may be interesting in its own right. Decompositions into stabbing-independent subsets may also be a combinatorial approach for getting approximation methods for stabbing problems.

Acknowledgments

We thank Joe Mitchell for pushing us to work on this problem, and repeated discussions that assured further progress. We also thank Kamal Jain for some discussions on iterated rounding.

References

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

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

[3] E.A. Arkin, M.E. Bender, E.D. Demaine, S.P. Fekete, J.S.B. Mitchell, and S. Sethia. Optimal covering tours with turn costs. In *Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 138–147, 2001.

[4] W.J. Cook, W.H. Cunningham, W.R. Pulleyblank, and A. Schrijver. *Combinatorial Optimization*. Wiley, New York, 1998.

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

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

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

[8] S. P. Fekete. On simple polygonalizations with optimal area. *Discrete and Computational Geometry*, 23:73–110, 2000.

[9] 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.

[10] M. Held, J.T. Klosowski, and J.S.B. Mitchell. Evaluation of collision detection methods for virtual reality flythroughs. In *Proceedings of the 7th Canadian Conference on Computational Geometry*, pages 205–210, 1995.

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

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

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

[14] A. Jüttner. On budgeted optimization problems. *Mathematics of Operations Research*, to appear.

[15] T. Király. EGRES list of open problems in combinatorial optimization. `http://www.cs.elte.hu/egres/problems/prob_22/index2.html`, 2002.

[16] 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.

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

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

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

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

[21] G. Reinelt. TSPLIB – A traveling salesman problem library. *ORSA J. Comput.*, 3(4):376–384, 1991.

[22] A. Schrijver. Personal communication, 2002.

[23] J.R. Shewchuk. Stabbing Delaunay tetrahedralizations. Unpublished Manuscript, 2002. Dept. Electrical Engineering and Computer Science, UC Berkeley, CA.

[24] M.M. Solomon. VRPTW benchmark problems. `http://w.cba.neu.edu/~msolomon/problems.htm`, 2003.

[25] E. Welzl. On spanning trees with low crossing numbers. In B. Monien and Th. Ottmann, editors, *Data Structures and Efficient Algorithms*, volume 594 of *Lecture Notes in Computer Science*, Berlin, 1992. Springer-Verlag.