

# The Minimum Stabbing Triangulation Problem: IP Models and Computational Evaluation

Breno Piva and Cid C. de Souza

University of Campinas, Institute of Computing, Av. Albert Einstein 1251, B.  
Geraldo, 13083-852 Campinas, São Paulo, Brasil.  
{bpiva, cid}@ic.unicamp.br

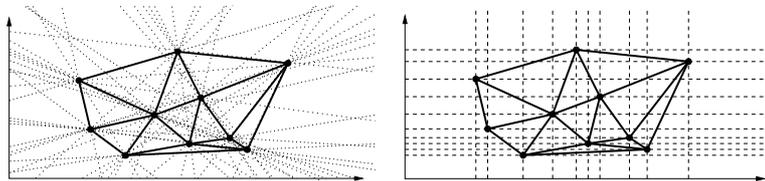
**Abstract.** The minimum stabbing triangulation of a set of points in the plane (MSTR) was previously investigated in the literature. The complexity of the MSTR remains open and, to our knowledge, no exact algorithm was proposed and no computational results were reported earlier in the literature of the problem. This paper presents integer programming (IP) formulations for the MSTR, that allow us to solve it exactly through IP branch-and-bound (B&B) algorithms. Moreover, one of these models is the basis for the development of a sophisticated Lagrangian heuristic for the problem. Computational tests were conducted with two instance classes comparing the performance of the latter algorithm against that of a standard (exact) B&B. The results reveal that the Lagrangian algorithm yielded solutions with minute, and often null, duality gaps for instances with several hundreds of points in small computation times.

## 1 Introduction

Given a set of points  $P$  in the plane, the *geometric graph* associated to  $P$  is the graph  $G(P) = (V, E)$  whose vertices are the points in  $P$  and whose edges are the straight line segments with both extremities in  $P$ . A *triangulation* of  $P$  is any *maximal* subset  $T$  of  $E$  such that no edge of  $T$  intersects another edge of  $T$ , except possibly in just one endpoint, or a point of  $P$  other than its own endpoints. In other words, any maximal planar subset of  $E$  defines a triangulation of  $P$ . When weights are associated to the edges in  $E$ , one can define the Minimum Weight Triangulation problem (MWT), in which the goal is to find a triangulation that minimizes the sum of all edge weights. The MWT occurs naturally in several contexts and for this reason has been studied for many years. For arbitrary weights, the problem was known for a long time to be  $\mathcal{NP}$ -hard. However, for the case in which the weights are given by the Euclidean distances, the complexity remained open until recently when Mulzer and Rote [12] proved that it is  $\mathcal{NP}$ -hard too.

In this paper we studied the Minimum Stabbing Triangulation problem (MSTR). The input of the MSTR comprises a set of points  $P$ , as in the MWT, and a set of straight lines  $L$  in the plane. Given a triangulation  $T$  of  $P$ , the *stabbing number* of  $T$  with respect to  $L$  is the maximum number of intersections between any line in  $L$  and the edges in  $T$ . The goal in the MSTR is to find a triangulation

with the smallest possible stabbing number. Two versions of the problem are presented in [7, 8] and are related to the choice of the set  $L$ . In the first version, here referred as the *general stabbing* one,  $L$  is defined as the infinite set formed by all straight lines that can be drawn in the plane. In the *axis parallel* version,  $L$  is the finite set composed solely by the vertical and horizontal lines associated to the coordinates of the points in  $P$ . Figure 1 illustrates the two versions of the problem with a triangulation of stabbing numbers 14 and 9, respectively. Before we continue, it is worth mentioning that in [7, 8] the authors proved that the closely related problem of finding a triangulation of minimum axis-parallel crossing number is  $\mathcal{NP}$ -hard. Due to space limitations, we do not discuss the latter problem here, restricting ourselves to call the reader's attention that the crossing and the stabbing numbers are different measures not to be confused with each other.



**Fig. 1.** A triangulation with general (axis parallel) stabbing number 14 (9).

**Motivation.** Stabbing problems can be defined more generally by replacing triangulation by any other geometric structure. Once the structure has been fixed, the goal is to find the one having the minimum stabbing number. Such problems have received considerably attention in the Computational Geometry community. In 2001 Mitchell and O'Rourke published a list with thirty open problems in the field [9], given rise to *The Open Problems Project* [5], containing a list of geometric problems whose complexity, at that time, was unknown. The list, which is constantly updated, is an invaluable source of challenging problems in Computational Geometry. Among those, problem #20, known as the Minimum Stabbing Spanning Tree problem, was investigated in [7, 8] and proved to be  $\mathcal{NP}$ -hard. In these works, the authors also shown that the Minimum Stabbing Matching is  $\mathcal{NP}$ -hard, developed an approximation algorithm and presented computational results. The MSTR was also introduced, but the complexity status of the problem could not be established and no algorithms were developed or tested to solve it. Spanning tree, matching and triangulation stabbing problems were investigated in [10] though, in the triangulation case, no guarantees were given for the quality of the solutions found. Other works related to finding structures with minimum or low stabbing number include [4], [1], [15] and [16].

**Our contribution.** This paper presents two IP formulations for the MSTR based on the ideas described in [7, 8, 13] and use them to implement an exact branch-

and-bound (B&B) algorithm, providing the first reported computational results for the problem in the literature. A Lagrangian relaxation (LR) heuristic based on the IP models is presented and a subgradient algorithm is implemented. Computational results obtained by this algorithm are reported with instances taken from the literature and reveal that optimality or minute duality gaps are achieved in small computation times.

To achieve these results, it was of paramount importance the realization of the relation existing between the MWT and the MSTR, which led to the development of strong IP models for the latter, and also the usage of efficient algorithms to solve the MWT, as these routines were part of our Lagrangian heuristic.

**Organization of the text.** The remaining of this document is organized as follows. Section 2 is devoted to the study of the MWT problem which as said above, is deeply related to the MSTR. In Sect. 3 two IP models for the MSTR are described. Section 4 describes how to derive a LR heuristic for the MSTR from the IP models, whilst in Sect. 5 we present our computational results. At last, in Sect. 6 we draw some conclusions and indicate future research directions to be pursued in our research.

## 2 The Minimum Weight Triangulation Problem

This section is devoted to the presentation of IP models for the MWT available in the literature and of a special structure, named LMT-skeleton [6], having the property of being a subset of an optimal solution to the problem. The LMT-skeleton can be computed efficiently and plays a central role in the LR heuristic proposed here.

**Integer programming models.** Two IP models were found for the MWT in the literature and are briefly discussed here. The first one will be called the *Edge Model* because each of its variables correspond to an edge connecting two points of the point set  $P$ . The second model, referred here as Triangle Model, for each variable corresponds to a, so-called, *empty* triangle, i.e., a triangle with vertices in  $P$  and no other point of  $P$  in its interior. These two models and computational results related to them are thoroughly discussed in [13]. Given  $P$ , its geometric graph  $G(P) = (V, E)$  and weight  $c_{ij}$  associated to each edge  $(i, j) \in E$ , the Edge Model of MWT defined over  $P$  is given below:

$$(MWTEB) \quad z = \min \sum_{ij \in E} c_{ij} x_{ij} \quad (1)$$

$$\text{subject to} \quad \sum_{ij \in E} x_{ij} = 3n - v - 3, \quad (2)$$

$$\sum_{ij \in Cr} x_{ij} \leq 1, \quad \forall \text{ crossing set } Cr \in G, \quad (3)$$

$$\sum_{ij \in Cr(kl)} x_{ij} \geq 1, \quad \forall kl \in E, \quad (4)$$

$$x_{ij} \in \mathbb{B}, \quad \forall ij \in E. \quad (5)$$

In this model, for every  $(i, j) \in E$ ,  $x_{ij} = 1$  if and only if the edge  $ij$  is in the triangulation. Defining  $n = |P|$  and  $v$  is the number of vertices in the convex hull of  $P$ , (2) guarantees that the solution will have the right number of edges. Now, if a *crossing set* is defined as a maximal set of edges which are pairwise intersecting (endpoints excluded), constraint (3) states that only one edge in such a set can be in the solution, thus, ensuring planarity. For every edge  $kl \in E$ , denote by  $Cr(kl)$  the set of edges intersecting  $kl$  (again with endpoints excluded) plus  $kl$  itself. Constraint (4) states that, either  $kl$  or at least one of the edges in  $Cr(kl)$  must be in the solution, therefore, enforcing maximality. It is noteworthy that constraint (4) is not strictly necessary for the formulation, however, as discussed in [13], it causes a great computational impact. Also, one must observe that, potentially, there are exponentially many constraints in (3). However as mentioned in [13], a valid and compact formulation can be obtained by adding one such constraint for each proper intersection between two edges in  $E$ . In this case, the model will have  $O(n^2)$  variables and  $O(n^4)$  constraints.

In the description of the Triangle Model below,  $\Delta(P)$  is the set of empty triangles over  $P$ ;  $L^+(ij)$  and  $L^-(ij)$  are the two half-planes defined by the line containing  $ij$ ;  $E_H$  is the set of edges in the convex hull of  $P$ :

$$(MWTTB) \tag{6}$$

$$\text{subject to } z = \min \sum_{ijk \in \Delta(P)} c_{ijk} x_{ijk} \tag{7}$$

$$\sum_{\substack{ijk \in \Delta(P) \\ ijk \subset L^+(ij)}} x_{ijk} = \sum_{\substack{ijk \in \Delta(P) \\ ijk \subset L^-(ij)}} x_{ijk}, \quad \forall ij \in E \setminus E_H, \tag{8}$$

$$\sum_{\substack{ijk \in \Delta(P) \\ ijk \supset ij}} x_{ijk} = 1, \quad \forall ij \in E_H, \tag{9}$$

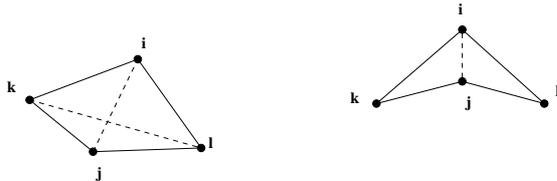
$$x_{ijk} \in \mathbb{B}, \quad \forall ijk \in \Delta(P). \tag{10}$$

In the Triangle Model, for every triangle  $ijk \in \Delta(P)$ ,  $x_{ijk} = 1$  if and only if the triangle  $ijk$  is in the triangulation. The weight of the triangle  $ijk$  is based on the weight of its edges and whether they are in the convex hull or not. It is defined by:  $c_{ijk} = \alpha_{ij}c_{ij} + \alpha_{ik}c_{ik} + \alpha_{jk}c_{jk}$ , where  $\alpha_{ij} = 1$  if  $ij$  is in the convex hull and  $\alpha_{ij} = 0.5$  otherwise. It is not hard to check that, with the values of  $\alpha$  taken in this way, the optimum of the Edge and Triangle models are the same. Constraint (8) states that the number of triangles containing an edge  $ij$  must be the same in both half-planes defined by the line containing  $ij$ . As the edges in the convex hull are present in every planar triangulation, constraint (9) ensures that a triangle containing one such edge is in the triangulation. One can check that this model has  $O(n^3)$  variables and  $O(n^2)$  constraints. Nunes [13] proved that the bound provided by the relaxation of the Triangle Model is not smaller than that of the Edge Model, i.e., from the IP viewpoint, it is a stronger formulation.

**The LMT-skeleton.** As the MWT is  $\mathcal{NP}$ -hard, there is little hope that an exact polynomial algorithm exists for this problem. Nevertheless, some polynomial

algorithms have been proposed to find non-trivial subsets of an optimal MWT solution. One of these subsets is given by the LMT-skeleton.

Given a planar triangulation  $T$  and an edge  $ij \in T$  but not in the convex hull,  $ij$  is a side of two empty triangles in  $T$ . These two triangles together make a quadrilateral  $ijkl$  having  $ij$  and  $kl$  as its diagonals. We say that  $ij$  is locally minimum if this quadrilateral is not convex or, else, if the weight of  $ij$  is smaller than the weight of  $kl$ . Figure 2 shows locally minimum segments. When all the edges



**Fig. 2.** Locally minimum segments. In both cases  $ij$  is locally minimum.

in a planar triangulation are locally minimum, we say that the triangulation itself is also locally minimum. Clearly, any minimum weight triangulation is locally minimum, however, not all locally minimum triangulations have minimum weight. The LMT-skeleton (short for Locally Minimum Triangulation Skeleton) is the subset of edges that are present in every locally minimum triangulation and, thus, is also a subset of any minimum weight triangulation.

In [6] the authors proposed a polynomial algorithm to find a LMT-skeleton and in [3] the algorithm was improved. The computational experiments performed with these algorithms showed that together with a dynamic programming algorithm to find a MWT of convex polygons it was possible to find the MWT of instances with thousands of points. The source code for this last algorithm written by Mulzer is available online at [11].

### 3 IP Models for the Minimum Stabbing Triangulation

The work in [7, 8] and the IP models for the MWT form the point of departure to build the Edge and Triangle Stabbing models for the MSTR. Again, we are given the sets  $P$  and  $L$  of points and stabbing lines, respectively, and  $E$  denotes the set of edges of the geometric graph  $G(P)$ . Then, the Edge Stabbing Model reads:

$$(MSTEB) \quad z = \min k \quad (11)$$

subject to (2),(3),(4),(5)

$$\sum_{ij \in E: ij \cap s \neq \emptyset} x_{ij} \leq k, \quad \forall s \in L. \quad (12)$$

In this formulation the variable  $k$  denotes the stabbing number and, therefore, has to be minimized. Constraint (12) states that, for each stabbing line  $s$  in  $L$ ,

the number of edges from triangulation that intersect  $s$  are bounded from above by the stabbing number. Notice that, as observed in [7, 8], in principle, this formulation is not finite since there are infinitely many stabbing lines. However, considering the axis parallel version, when sweeping a stabbing line in a direction  $d$ , the stabbing number only changes at a point of  $P$ . For this reason, we only need to look at a linear number of stabbing lines, thus, making the model finite. Following a similar reasoning, when considering the general version, we only need to look at a quadratic number of lines.

As for the Triangle Stabbing Model, it reads:

$$(MSTTB) \quad z = \min k \tag{13}$$

$$\text{subject to} \quad (8),(9),(10),$$

$$\sum_{ijl \in \Delta(P):ijl \cap s \neq \emptyset} c_{ijl} x_{ijl} \leq k, \quad \forall s \in L. \tag{14}$$

As in the previous case, the first constraints are the same as in the Triangle Model for the MWT and guarantee that the vector  $x$  represents a triangulation of  $P$ . Moreover,  $k$  is defined again as the stabbing number and, thus, is minimized in the objective function. Constraint (14) states that the sum of the coefficients  $c_{ijl}$  of the triangles  $ijl$  intersecting a line  $s$  of  $L$  can not be larger than the stabbing number. A triangle  $ijl$  intersecting a line  $s$  has coefficient  $c_{ijl} = \beta_{ij} + \beta_{il} + \beta_{jl}$ , where  $\beta_{ij} = 1$  if  $ij$  intersects  $s$  and is on the convex hull,  $\beta_{ij} = 0.5$  if  $ij$  intersects  $s$  but is not on the convex hull and  $\beta_{ij} = 0$  if  $ij$  does not intersect  $s$ .

## 4 A Lagrangian Relaxation for the MSTR

Using the IP formulations from the previous section, we now derive a Lagrangian relaxation (LR) model for the MSTR. We solve the dual of this relaxation via the subgradient method (SGM), which allows us to obtain a lower bound for the optimal value of the MSTR. Besides, at each iteration of the SGM, we compute the primal Lagrangian problem whose solution is a triangulation and, thus, can be used to obtain upper bounds for the MSTR. For the basic theory of Lagrangian relaxation the reader is referred to [17].

The presentation of our LR is based on the Edge Stabbing Model described in Sect. 3. However, it is straightforward to apply the same ideas to the Triangle Stabbing Model. To obtain the LR ( $MSTEB(u)$ ) of problem ( $MSTEB$ ) we simply dualize the constraints (12), penalizing them in the objective function value. This operation results in the following model:

$$(MSTEB(u)) \quad z = \min k - \sum_{s \in L} u_s (k - \sum_{ij \in E:ij \cap s \neq \emptyset} x_{ij}) \tag{15}$$

$$\text{subject to} \quad (2),(3),(4),(5)$$

Notice that this formulation is equivalent to the formulation for the MWT discussed in Sect. 2, with the weight  $c_{ij}$  of edge  $ij$  given by:

$$c_{ij} = \sum_{s \in L:s \cap ij \neq \emptyset} u_s. \tag{16}$$

At first glance, since the MWT is  $\mathcal{NP}$ -hard, one may wonder if this relaxation is useful after all. However, as observed in Sect. 2, subsets of optimal solutions of the problem like the LMT-skeleton can be computed efficiently. In practice, one can expect to solve instances of the MWT with several hundreds of points very quickly. Our approach relies on this observation and the results reported in Sect. 5 confirmed our expectations.

Now, as  $(MSTEB(u))$  is a relaxation of  $(MSTEB)$ , we know that  $z(u) \leq z$  and, since we want to find the best possible bound, we must find the value of  $u$  that maximizes  $z(u)$ , i.e., we must solve the dual Lagrangian problem given by:

$$(DL) \quad v_{DL} = \max\{z(u) : u \geq 0\}. \quad (17)$$

The problem  $(DL)$  can be solved using the SGM as described in [17] and [2]. To this end, the multipliers  $u_s$  are initialized with null values and are updated at iteration  $t$  by the formula:

$$u_s^t = \max(0, u_s^{t-1} - \mu G_s^{t-1}). \quad (18)$$

with  $\mu$  given by

$$\mu = \frac{\pi(\text{dist} \times ub - lb)}{\sum_{s \in L} (G_s^{t-1})^2}, \quad (19)$$

and  $G_s^{t-1}$ , the  $s$ -th component of the subgradient of  $z(u)$  in  $u^{t-1}$ , given by

$$G_s^{t-1} = k - \sum_{ij:ij \cap s \neq \emptyset} x(u^{t-1})_{ij}. \quad (20)$$

In the formulas above,  $ub$  and  $lb$  are, respectively, an upper and a lower bound for the optimal value,  $\text{dist}$  is a perturbation factor (arbitrarily set to 1,05 in our experiments) and  $\pi$  is the step size (in our experiments initialized at 2 and halved every 30 iterations without improvement in the lower bound). The solution of the Lagrangian primal problem is denoted by  $x(u)$  and the superscripts indicate the iteration at which each variable is being considered (e.g.,  $u^t$  is the Lagrangian multipliers vector at iteration  $t$ ).

Now, notice that, after dualizing the constraints (12), the objective function of  $(MSTEB(u))$  can be rewritten as:

$$z(u) = \min k(1 - \sum_{s \in L} u_s) + \sum_{ij \in E} x_{ij} \sum_{s \in L: s \cap ij \neq \emptyset} u_s. \quad (21)$$

Therefore, if  $\sum_{l(d) \in d} u_{l(d)} > 1$ , the first term of that equation would have a negative value and, the larger the value of  $k$ , the smaller the value of  $z(u)$ . As a result, when optimizing the (primal) Lagrangian problem, if the cost of variable  $k$  is negative, the lower bound  $z(u)$  is unlimited and hence useless. Analogously, if the cost of  $k$  is non negative, the obvious solution is to set  $k$  to zero. However, by doing so, we may waste the opportunity to produce a better dual bound for  $z$ . To overcome these situations, we proceed in the following way. In the solution of

( $MSTEB(u)$ ),  $k$  is set, respectively, to the best upper ( $ub$ ) or lower ( $lb$ ) bound available for  $z$  depending on whether its cost is negative or not. In fact, in our implementation, when the cost is non negative,  $k$  is set to  $\lceil lb \rceil / 2$  rather than to  $lb$  to avoid an early convergence of the SGM. This tends to increase the number of iterations of the method, augmenting the chances of the Lagrangian heuristic to obtain a better feasible solution.

The termination criteria implemented in our SGM are achieved when one of the following situations occur: the difference between the upper and lower bounds is smaller than 1 (one), the value of  $\pi$  is smaller than 0,005, or yet, a predefined time limit is reached.

**Lagrangian Heuristic.** Each iteration of the SGM solves a MWT. The solution of this problem is a triangulation of  $P$  and, therefore is also feasible for the MSTR. So, an upper bound for the optimal value of the MSTR can be immediately obtained by computing the stabbing number of this triangulation.

**Solving the Lagrangian Primal.** The MWT corresponding to ( $MSTEB(u)$ ) is solved in our implementation through the following steps. First we determine three subsets  $T_m$ ,  $T_p$  and  $T_f$  of edges which, respectively, are mandatory, possible and forbidden in a optimal solution, using a LMT-skeleton algorithm. Then, we are left with a constrained MWT problem where all edges of  $T_m$  are forced to be in the solution, the ones in  $T_f$  are eliminated from the solution and those in  $T_p$  are uncertain. Typically, this dramatically reduces the set of edges that remain undecided. Thus, we can make use of the Edge Model from Sect. 2 with the appropriate variables set to one or zero or, the Triangle Model with the variables corresponding to triangles containing edges in  $T_f$  set to zero and use an IP solver to compute the model via a standard branch-and-bound algorithm. As we will see later in this paper, it turns out that this procedure is capable to solve the Lagrangian problems in an extremely efficient fashion.

## 5 Computational Results

We now describe the experiments we carried out to test the performance of the algorithms discussed in the previous sections. As mentioned earlier, we implemented exact B&B algorithms based on the IP models discussed in Sect. 3 as well as the LR algorithms from Sect. 4.

The first stage of our testing comprised a comparison of the two alternative B&B algorithms that arise from the Edge and Triangle stabbing models discussed in Sect. 3. For the MWT, it was observed in [13] that the B&B algorithm performs better when it uses the Triangle Model rather than the Edge Model. Hence, a similar behavior was expected from the corresponding models when applied to the solution of the MSTR. Indeed, this was what happened and, thus, all the B&B results reported below were obtained using the Triangle Stabbing Model.

As for the LR algorithm, we implemented the subgradient algorithm using the Edge Stabbing Model instead of the Triangle Stabbing Model. The reason for doing this is simply that in this case, relaxing the stabbing constraints in

both models result in a MWT problem, the only difference been the equation used to calculate the edge weights. Thus, as the equation is simpler in the Edge Stabbing Model, this model was chosen.

As said in Sect. 4, to solve the Lagrangian primal problem, we used the LMT-skeleton code available in [11] and written by Mulzer following the description in [3]. A few modifications were introduced in this program to make possible the usage of arbitrary edge weights instead of Euclidean ones. This included, for instance, the removal of the *diamond test*, a simple and effective way to determine whether an edge could be part of a triangulation of minimum (Euclidean) length. Such changes do not have significantly damaged the algorithm’s performance, relative to Euclidean weights, confirming it as a viable option for general MWTs.

After running the LMT-skeleton, quite often we still do not have a triangulation, and a B&B algorithm is used to solve the remaining problem, i.e., a MWT with sets of mandatory and forbidden edges. Since we use the Triangle Model as the input for the B&B algorithm, these sets of edges have to be processed to identify the corresponding sets of triangles. Thus, if an empty triangle contains a forbidden edge, the associated variable is set to zero while, if all the edges forming its sides are mandatory, this variable is set to one.

**Computational Environment.** To perform the experiments, we used a computer with an Intel Core 2 Quad 2.40GHz, 4096 KB cache, 4GB of RAM memory and a Ubuntu 8.10 OS. The programming language used was C/C++ with `gcc` 4.3 compiler and every program was compiled with `-O5` optimization flag. We also used the `XPRESS-Optimizer 64-bit v20.00.05` IP solver. The default cuts, heuristics and preprocessing were turned off.

**Instances.** As a test suite we used 26 instances from TSPLIB [14] and 25 regular grid instances used in [7] ( $5 \times 5$  to  $20 \times 20$  grids with 20% of its points randomly removed) for the Minimum Stabbing Matching Problem. The choice of these instances is based on the fact that the TSPLIB is a well known test library for geometric problems and, besides, some TSPLIB and all grid instances were used in [7] for a related problem. The choice of the instance sizes was made seeking tests that were hard enough to provide sufficiently high computation times in order to allow a more precise comparison of the algorithms.

In every test, a time limit of 1,800 seconds was set for the computations. Notice, however, that in the tables showed below, occasionally the time is slightly bigger than this limit. This happens because the time limit is verified at certain points in the program codes and, it could be that the time elapsed between two such checks is not negligible. This situation arises, for example, when the model of a big instance is being uploaded by the IP solver. In our experiments an additional timeout script running on the operating system level was used that forces the process to halt after 2,000 seconds. In case the process ends naturally, a bound is always produced. On the other hand, if the process is killed by the timeout script, no output is produced. The latter situation is signaled in the tables by the symbol ‡.

**Analysis of the results.** We divide our analysis into two parts, one for the TSP instances and the other for the grid instances. For both algorithms we tested,

duality gaps were computed through the formula  $100 \times (ub - lb) / lb$ , where  $ub$  and  $lb$  denote, respectively, the upper and lower bounds yielded by the algorithm. Concerning the TSP instances, the B&B algorithm had its process killed in 13 out of the 26 instances and, when this was not the case, it proved optimality in all but one instance (kroC100), where there is a 3.45% gap. On the other hand, the Lagrangian SGM converged in all cases within the imposed time limit, with an average gap of 2.98%. The performance of the heuristic is remarkable. Optimality was proven for 8 instances, two of which could not be reached by the exact algorithm within the time limit (the inverse situation occurred six times). In 13 instances the difference between the upper and lower bounds was of just one unit. The results suggested that better performances could be achieved by improving the primal heuristic, since the lower bound provided by SGM reached the optimum in 14 cases. The computation times of the Lagrangian heuristic were usually much smaller than that of the B&B algorithm. However, this sort of comparison is not really adequate since the B&B is an exact algorithm while LR is just a heuristic. So, we just observe that, constraining ourselves to the 6 instances solved to optimality by both algorithms, B&B was faster in just one of them (pr152) and was more than a 100 times slower in another (rd100).

The results for the grid instances can be seen in Table 2. For those instances, the Lagrangian subgradient algorithm was able to solve to optimality every instance but one (grid20c), for which there is a gap of 1.12% (an absolute gap of only one unit). The B&B algorithm was unable to solve 5 out of 25 grid instances. In fact, none of the  $20 \times 20$  grid instances was solved within the time limit and every other grid instance was solved to optimality. Again, one has to be careful when comparing execution times since the algorithms are of different nature. Bearing this in mind and constraining our analysis to the  $g \times g$  grid instances with  $g \leq 15$  (the ones for which both algorithms proved optimality), we just notice that the B&B algorithm had a slightly better performance when  $g \leq 10$ . For  $g = 15$ , i.e., for the larger instances, this situation is reversed and the Lagrangian method performs much better.

## 6 Conclusions and Future Directions

To our knowledge, in this paper we proposed the first exact approach to tackle the MSTR. Moreover, we also devised a Lagrangian heuristic for the problem and conducted several computational experiments with it, which showed that it yields solutions of high quality, often of proven optimality, in small computation times. We are not aware of another work in the literature which reports on computational results for this problem.

Future directions in this research are currently being considered. This includes the usage of Lagrangian relaxation to tackle other stabbing problems such as those related to spanning trees and matchings. We also consider to improve the performance of our heuristic for the MSTR by adding new features to it, such as, a procedure for variable fixing in the traditional Lagrangian fashion and a fast local search to reduce the primal bounds.

Table 1: Results for TSP instances.

Instance	Time		LB		UB		GAP%	
	LR	B&B	LR	B&B	LR	B&B	LR	B&B
berlin52	2.93	6.02	24	24	24	24	0.00	0.00
ch130	154.97	‡	32	‡	33	‡	3.12	‡
ch150	225.25	‡	34	‡	35	‡	2.94	‡
d198	1,707.69	1,081.32	56	56	56	56	0.00	0.00
eil76	25.89	32.21	32	32	33	32	3.12	0.00
gil262	1,301.22	‡	49	‡	50	‡	2.04	‡
gr202	650.82	‡	42	‡	42	‡	0.00	‡
kroA100	72.85	667.80	29	29	30	29	3.45	0.00
kroA150	297.60	‡	35	‡	36	‡	2.86	‡
kroA200	885.11	‡	40	‡	41	‡	2.50	‡
kroB100	66.17	1,065.68	29	29	30	29	3.45	0.00
kroB150	292.88	‡	34	‡	35	‡	2.94	‡
kroB200	499.98	‡	39	‡	40	‡	2.56	‡
kroC100	48.50	1,877.93	29	29	29	30	0.00	3.45
kroD100	20.42	1,182.55	29	29	29	29	0.00	0.00
kroE100	78.48	651.14	29	29	30	29	3.45	0.00
lin318	1,800.05	‡	69	‡	71	‡	2.90	‡
pr124	466.37	189.36	48	49	49	49	2.08	0.00
pr136	700.72	148.89	66	66	67	66	1.52	0.00
pr144	745.97	259.06	74	74	74	74	0.00	0.00
pr152	443.70	996.30	45	45	45	45	0.00	0.00
pr226	1,812.37	‡	140	‡	150	‡	7.14	‡
pr264	1,811.61	‡	90	‡	92	‡	2.22	‡
rd100	16.03	1,683.79	29	29	29	29	0.00	0.00
rd400	1,803.94	‡	52	‡	55	‡	5.77	‡
u574	1,818.41	‡	73	‡	90	‡	23.2	‡

Table 2: Results for grid instances.

Instance	Time		LB		UB		GAP%	
	LR	B&B	LR	B&B	LR	B&B	LR	B&B
grid5a	0.04	0.02	22	22	22	22	0.00	0.00
grid5b	0.05	0.02	21	21	21	21	0.00	0.00
grid5c	0.03	0.02	21	21	21	21	0.00	0.00
grid5d	5.13	0.02	21	21	21	21	0.00	0.00
grid5e	0.03	0.02	20	20	20	20	0.00	0.00
grid8a	0.86	0.24	34	34	34	34	0.00	0.00
grid8b	1.56	0.42	34	34	34	34	0.00	0.00
grid8c	0.82	0.18	34	34	34	34	0.00	0.00
grid8d	0.41	0.20	35	35	35	35	0.00	0.00
grid8e	0.47	0.95	35	35	35	35	0.00	0.00
grid10a	5.66	1.17	44	44	44	44	0.00	0.00
grid10b	2.33	0.76	42	42	42	42	0.00	0.00
grid10c	7.56	1.55	47	47	47	47	0.00	0.00
grid10d	1.41	0.82	46	46	46	46	0.00	0.00
grid10e	5.16	1.61	46	46	46	46	0.00	0.00
grid15a	73.08	212.52	66	66	66	66	0.00	0.00
grid15b	13.66	312.10	68	68	68	68	0.00	0.00
grid15c	16.67	190.91	64	64	64	64	0.00	0.00
grid15d	50.96	141.11	66	66	66	66	0.00	0.00
grid15e	62.80	238.79	67	67	67	67	0.00	0.00
grid20a	640.38	‡	89	‡	89	‡	0.00	‡
grid20b	86.33	‡	86	‡	86	‡	0.00	‡
grid20c	1,813.74	‡	89	‡	90	‡	1.12	‡
grid20d	252.37	‡	87	‡	87	‡	0.00	‡
grid20e	1,356.85	‡	90	‡	90	‡	0.00	‡

**Acknowledgments.** This research was partially supported by *Conselho Nacional de Desenvolvimento Científico e Tecnológico* – grants #301732/2007-8, #473867/2010-9, #147619/2010-6; *Fundação de Amparo à Pesquisa do Estado de São Paulo* – grant #07/52015-0, and a grant from FAEPEX/UNICAMP. The authors are also grateful to Prof. M. Lübbecke for making available the grid instances.

## References

1. P. Agarwal, B. Aronov, and S. Suri. Stabbing triangulations by lines in 3d. In *Proceedings of the eleventh annual symposium on Computational geometry*, SCG '95, pages 267–276, New York, NY, USA, 1995. ACM.
2. J. Beasley. Lagrangean relaxation. In *Modern Heuristic Techniques for Combinatorial Problems*, pages 243–303. McGraw-Hill, 1993.
3. R. Beirouti and J. Snoeyink. Implementations of the LMT heuristic for minimum weight triangulation. In *Proceedings of the fourteenth annual symposium on Computational geometry*, SCG '98, pages 96–105, New York, NY, USA, 1998. ACM.
4. M. Berg and M. Kreveld. Rectilinear decompositions with low stabbing number. *Information Processing Letters*, 52(4):215 – 221, 1994.
5. E. Demaine, J. Mitchell, and J. O'Rourke. The open problems project. Available online (accessed in January 2010). <http://maven.smith.edu/~orourke/TOPP/>.
6. M.T. Dickerson and M.H. Montague. A (usually) connected subgraph of the minimum weight triangulation. In *Proc. of the 12th Annual ACM Symp. on Comp. Geom.*, pages 204–213, 1996.
7. S. Fekete, M. Lübbecke, and H. Meijer. Minimizing the stabbing number of matchings, trees, and triangulations. In J. Munro, editor, *SODA*, pages 437–446. SIAM, 2004.
8. S. Fekete, M. Lübbecke, and H. Meijer. Minimizing the stabbing number of matchings, trees, and triangulations. *Disc. Comp. Geometry*, 40:595–621, 2008.
9. J. Mitchell and J. O'Rourke. Computational geometry. *SIGACT News*, 32(3):63–72, 2001.
10. J. Mitchell and E. Packer. Computing geometric structures of low stabbing number in the plane. In *Proc. 17th Annual Fall Work. on Comp. Geom. and Vis.* IBM Watson, 2007.
11. W. Mulzer. LMT-skeleton program. Available online (accessed in March 2011). [http://page.mi.fu-berlin.de/mulzer/pubs/mwt\\_software/old/ipelets/LMTSkeleton.tar.gz](http://page.mi.fu-berlin.de/mulzer/pubs/mwt_software/old/ipelets/LMTSkeleton.tar.gz).
12. W. Mulzer and G. Rote. Minimum-weight triangulation is NP-hard. *Journal of the ACM*, 55:11:1–11:29, 2008.
13. A. P. Nunes. Uma abordagem de programação inteira para o problema da triangulação de custo mínimo. Master's thesis, Institute of Computing, University of Campinas, Campinas, 1997. In Portuguese.
14. G. Reinelt. TSPLIB. Available online (accessed in March 2011). <http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/>.
15. J. R. Shewchuk. Stabbing Delaunay tetrahedralizations. *Disc. Comp. Geometry*, 32:343, 2002.
16. C. D. Tóth. Orthogonal subdivisions with low stabbing numbers. volume 3608 of *Lect. Not. in Comp. Science*, pages 256–268. Springer Berlin / Heidelberg, 2005.
17. L. A. Wolsey. *Integer Programming*. John Wiley and Sons, 1998.