Metaheuristic approaches for the Minimum Dilation Triangulation problem

Maria Gisela Dorzán*[1], Mario Guillermo Leguizamón†[1], Efrén Mezura-Montes‡[2], and Gregorio Hernández§[3]

[1]Universidad Nacional de San Luis - Argentina
[2]Universidad Veracruzana - México
[3]Universidad Politécnica de Madrid - España

Abstract

We focus on the development of approximated algorithms to find high quality triangulations of minimum dilation because the complexity status of the Minimum Dilation Triangulation problem for a general point set is unknown. We propose an operator to generate the neigborhood which is used in different algorithms: Local Search, Iterated Local Search, and Simulated Annealing. Besides, an algorithm called Random Local Search is presented where good and bad solutions are accepted using the previous mentioned operator. We use the Sequential Parameter Optimization method for tuning the parameters of the SA algorithm. We compare our results with the only available algorithm found in the literature that uses the obstacle value to sort the edges in the constructive process. Through the experimental evaluation and statistical analysis, we assess the performance of the proposed algorithms using this operator.

Introduction

Different measures are adopted to design optimal triangulations. The most popular are the weight, stabbing number, area, and dilation. Although the usage of approximated approaches to solve complex optimization problems is common nowadays, this last measure has not been used as selection criterion when optimizing triangulations by using metaheuristic algorithms [10].

Let S be a finite planar point set, T a triangulation of S and u, v two points in S. There are two distance metrics: the Euclidean distance between u and v, $|uv|$, and the length of the shortest path between u and v with in the triangulation T, $dist_T(u,v)$. The dilation between u and v with respect to T is the ratio between the shortest path and the Euclidean distances between u and v, and is defined as $\Delta_T(u,v) = \frac{dist(u,v)}{|uv|}$.

The maximum over all the dilations between pairs of points in T is called the dilation of T (or stretch factor) and is represented by $\Delta(T)$. The best possible dilation of any triangulation of S is the dilation of S and is denoted by $\Delta(S)$. Thus, we have

$$\Delta(T) = \max_{u,v \in S} \Delta_T(u,v) \text{ and } \Delta(S) = \min_{T of S} \Delta(T)$$

The triangulation T^* whose dilation is the dilation of S, i.e. $\Delta(T^*) = \Delta(S)$, is called minimum dilation triangulation of S. Note that there are several triangulations of minimum dilation for a given set of points.

Computing the minimum dilation triangulation for a set of points in the plane is listed as an Open Problem in Eppstein's survey [6], i.e., no polynomial algorithm that can build it is known and no proof that the problem is NP-hard is shown. Neither the greedy nor the delaunay triangulation algorithms generate the minimum dilation triangulation of a planar point set, as shown in Figure 1. Therefore, one approach is to use metaheuristic techniques for obtaining approximate solutions to the optimum.

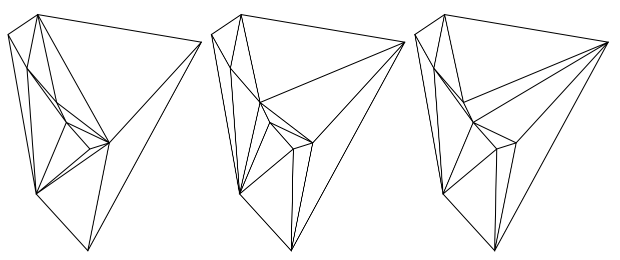
(a) GT (b) DT (c) MDT

Figure 1: A point set for which the MDT neither is the GT nor the DT.

*Email: mgdorzan@unsl.edu.ar. Research supported by CONICET and Research Project No. 22/F014

†Email: legui@unsl.edu.ar. Research supported by Laboratorio de Investigación y Desarrollo en Inteligencia Computacional (LIDIC)

‡Email: emezura@uv.mx. Research supported by CONACyT bilateral project No. 164626.

§Email: gregorio@fi.upm.es. Research supported by ESF EUROCORES programme EuroGIGA - ComPoSe IP04 - MICINN Project EUI-EURC-2011-4306

1 Approximated algorithms for the MDT problem

A set of simple techniques is presented because, to the best of the authors' knowledge, there are no works in the literature where the MDT problem is approached using metaheuristics. We present the general overview of the studied algorithms: Greedy [5], Local Search [1, 11], Iterated Local Search [9], Simulated Annealing [3, 7], and Random Local Search describing their features and parameters. For all the algorithms the solution space $\mathcal{E}$ is represented by all possible triangulations of a set S of n points in the plane. An $n \times n$ matrix of ones and zeros is used to represent a possible solution. A 1 at position (i, j) means that there is an edge connecting points i and j, otherwise a 0 is placed. The objective function, $f : \mathcal{E} \to R$, assigns a real value to each element of $\mathcal{E}$. For each $E \in \mathcal{E}$, the function f is defined as the maximum dilation among all pairs of points in E.

Greedy Algorithm (*G-MDT*) It starts with an empty solution *Sol* and inserts edges until a triangulation for a given set of point S is generated. Let A be the set of all possible edges that have not been inserted in *Sol*, $u, v \in S$ and $e = uv \in A$. If the dilation of the points u and v determines the dilation of the solution, i.e., $\Delta_{Sol}(u, v) = \Delta(Sol)$, then the edge e is inserted in the solution *Sol*. This reduces the dilation of the points u and v to 1 because $dist(u, v) = |uv|$ and consequently decreases the dilation of the solution *Sol*. The insertion is performed whenever the new edge to insert produces no intersections with the edges already inserted.

Local Search Algorithm (*LS-MDT*) It starts from a random initial solution and then iteratively moves to a neighbor solution. These moves are performed by applying an operator looking for a better solution. If a better solution is found, it replaces the current solution by the new one and it continues the process until it can not improve the current solution. When no improved solutions are present in the neighborhood, local search is stuck at a local optimal solution. The moves in the search space are performed using the *retriang()* operator that works as follows. Let $a, b \in S$, if the dilation of the points a and b determines the dilation of the current solution *Sol*, i.e., $\Delta_{Sol}(a, b) = \Delta(Sol)$ (see Figure 2(a)), then a and b are joined by an edge. The egdes intersected by ab are deleted and the region delimited by these edges (see Figure 2(b)) is retriangulated in a greedy way. The edge whose points have the higher dilation is inserted at each step if it does not intersect with the edges previously added and if a complete triangulation is not achieved (see Figure 2(c)). Therefore the current solution is improved because its dilation has been reduced. Due to the behavior of this operator only a single neighbor is obtained.

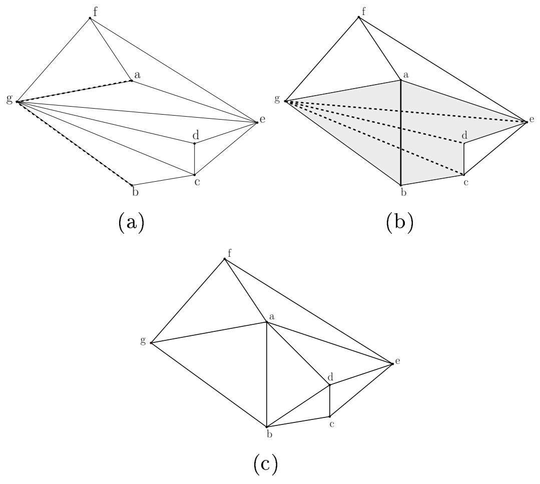

Figure 2: *retriang()* operator. In (a), $\Delta_{Sol}(a, b) = \Delta(Sol)$ is shown in dashed style. In (b), the edge ab is added and the dashed edges intersected by ab are deleted. The grey region is retriangulated in a greedy way (c).

Iterated Local Search Algorithm (*ILS-MDT*) First, a local search is applied to an initial random solution and yields a local optimum using the *LS-MDT* algorithm. At each iteration, a perturbation of *Sol* is carried out and then, a local search is applied to the perturbed solution *Sol'*. This process iterates until the number of perturbations is less than 50 and the current solution is improved, always keeping the best solution found so far Sol_{best}. The *perturb(Sol)* procedure is performed by the perturbation operator where $n/5$ random local retriangulations over random selected points of S are performed worsening the current solution, i.e., a point $b \in S$ is randomly chosen and all the points adjacent to b form the grey region that has to be retriangulated. The edges belonging to the region are deleted and then this region is retriangulated in a random way. The edges of the region are inserted at random if they do not intersect with the edges previously added and if a complete triangulation is not achieved.

Simulated Annealing Algorithm (*SA-MDT*) From an initial solution, it executes a number of iterations where a neighbor of the current solution *Sol* is generated in each one of them. The moves that improve the cost function are always accepted. Otherwise, the neighbor *Sol'* is selected with a given probability that depends on the current temperature T. This probability is usually called *acceptance function* and it is evaluated according to $p(T, Sol, Sol') = e^{-\frac{\delta}{T}}$

where $\delta = f(Sol') - f(Sol)$. $M(T_k)$ moves are performed for each temperature T_k. Finally, the value of T_k is decreased at each algorithm iteration k. The algorithm continues this way until the termination condition is met.

We use two operators to obtain the neighborhood of a solution: 1) Local random retriangulation (RR) as used in the perturbation operator in *ILS-MDT* algorithm; 2) Local greedy retriangulation (GR) is similar to the RR described above, but the edges are inserted in a greedy way. A point $b \in S$ is randomly chosen and all the points adjacent to b are recovered. Then the grey region that forms the adjacent points to b is retriangulated using at each step the edges whose points have the higher dilation (if it does not intersect with those previously added). The scheduling of application for these operators is as follows. GR is performed g times and RR is performed r times. We considered $g \in \{4, 6, 8\}$ and $r = 2$ as will be described later in the next section.

Due to the complexity involved in tuning the parameters of metaheuristic techniques, we used Sequential Parameter Optimization (SPO) [2] for tuning the parameters required by SA. All SPO-tuning experiments for the SA-MDT algorithm were performed with the following settings: 50 sequence steps, 5 new design points in each step, up to 2 repeats per design point, and 7 initial design points. Random Forest was used as a fast surrogate model building tool. Latin hypercube sampling was chosen as the generator of design points. We obtained better results with T_k moves at each temperature ($M(T_k) = T_k$), using local retriangulation interleaving with $g = 8$ and $r = 2$ and with a initial temperature equal the maximum length of the paths in the initial solution.

Random Local Search Algorithm (*RLS-MDT*) Initially, the proposed SA algorithm used the operator *retriang()* to move in the solution space. We observed that such algorithm converged too early to suboptimal regions (the best solution was achieved during the first temperature value). Therefore we proposed another operator for the SA algorithm to avoid this condition of premature convergence. On the other hand, the results obtained with operator *retriang()* were encouraging with the LS and ILS algorithms. Therefore, we propose a new algorithm; we called it Random Local Search algorithm, which starts with a random initial solution. $maxEvals = 2n$ evaluations are performed, accepting good and bad solutions. This mechanism allows to explore and exploit the search space, always keeping the best solution found so far Sol_{best}. We consider the operator *retriang()* but the retriangulation of the region is calculated in a random manner because better results were obtained that way.

2 Experimental Evaluation and Statistical Analysis

We generated 20 problem instances, each one is called n-i, where $n = \{40, 80, 120, 160, 200\}$ is the size of the instance and i is the number of the instance ($1 \leq i \leq 4$). The points are randomly generated, uniformly distributed and for each point (x, y), the coordinates $x, y \in [0, 1000]$. For stochastic algorithms 25 runs were carried out with different initial seeds and different initial triangulations for each run.

Table 1 shows the best values obtained with the proposed algorithms: *G-MDT*, *LS-MDT*, *ILS-MDT*, *SA-MDT*, and *RLS-MDT*. The lowest dilations are bolded. Xia proved that the dilation of the Delaunay triangulation of a set of points in the plane is less than 1.998 [12]. All the results obtained by the algorithms are less than 1.998 and the Delaunay triangulation never obtains better results for the instances considered (see the "*DT*" column). We compared our results with those obtained with the algorithm proposed in [8]. We called it *OV-MDT* because it computes the Obstacle Value of the edges for approximating the minimum dilation triangulation. We used the Java-applet available at the Geometry Lab site at the University of Bonn (www.geometrylab.de) to obtain the results. For some instances the applet did not produce a result giving an error and halting the execution of the algorithm. It should be noted that the applet yielded results after several days of execution (see the "*OV-MDT*" column). The *G-MDT* algorithm obtained solutions of poor quality even for the smallest instances. The *RLS-MDT* algorithm obtained the lowest (best) dilations for all problem instances (except for one instance of 120 points). We observed that the *OV-MDT* algorithm did not obtain better results than *RLS-MDT*. Although in some instances equal best values were obtained by some algorithms, the *RLS-MDT* algorithm showed to be less complex and faster than the other algorithms.

As all samples had a non-Normal distribution (Kolmogorov-Smirnov test) we used non-parametric statistical tests to evaluate the confidence on the statistical results provided by the algorithms. In order to carry out a comparison which involved results from more than two algorithms Kruskal Wallis test was used for multiple comparisons with independent samples [4]. We used the post-hoc Tukey test to find which algorithms' results are significantly different from one another. These tests are well-known and they are usually included in standard statistics packages (such as *Matlab*, *R*, etc.). After applying the Kruskal Wallis test, we observed that there was always a significant difference between the algorithms ($1.e\text{-}16 < p$-value ≤ 0.0015) being this difference much larger when considering sets with more points. The *ILS-MDT* and *RLS-MDT* algorithms had similar per-

Instance	*DT*	*OV-MDT*	*G-MDT*	*LS-MDT*	*ILS-MDT*	*SA-MDT*	*RLS-MDT*
40-1	1.31871	132.863	1.37203	130.959	**1.29467**	**1.29467**	**1.29467**
40-2	1.37491	**1.36881**	1.37491	**1.36881**	**1.36881**	**1.36881**	**1.36881**
40-3	**1.32268**	**1.32268**	1.36026	**1.32268**	**1.32268**	**1.32268**	**1.32268**
40-4	1.35391	**1.32330**	1.34919	**1.32330**	**1.32330**	1.32557	**1.32330**
80-1	1.33034	**1.32363**	1.39394	**1.32363**	**1.32363**	1.40844	**1.32363**
80-2	1.40500	135.181	1.38199	1.35339	**1.32418**	1.50331	**1.32418**
80-3	1.37791	**1.30519**	1.36180	1.34065	1.31457	1.37791	**1.30519**
80-4	1.42547	134.239	1.39362	1.33176	**1.31583**	1.47561	**1.31583**
120-1	1.37428	**1.34442**	1.42386	1.35398	1.34825	1.72082	**1.34442**
120-2	1.33973	131.194	1.37778	1.31194	**1.30366**	1.65921	1.31194
120-3	1.37724	134.016	1.43027	1.40314	1.31985	1.74912	**1.29786**
120-4	1.38529	**1.33600**	1.39972	1.35152	1.34272	1.68163	**1.33600**
160-1	1.34365	**1.31050**	1.43076	1.35236	1.33384	1.95530	**1.31050**
160-2	1.35409	*NA*	**1.33260**	1.36463	**1.33260**	1.97896	**1.33260**
160-3	1.35797	133.278	1.37723	1.37178	1.36352	1.87574	**1.32995**
160-4	1.37922	**1.34941**	1.37922	1.37922	1.35037	1.87033	**1.34941**
200-1	**1.35751**	*NA*	1.42220	1.41867	**1.35751**	2.08064	**1.35751**
200-2	1.39512	*NA*	1.39512	1.36451	**1.36350**	2.06324	**1.36350**
200-3	141.962	*NA*	1.45763	1.40645	**1.40645**	2.09818	**1.40645**
200-4	1.36965	*NA*	1.40765	1.35499	**1.35251**	2.00110	**1.35251**

Table 1: Best results of *DT*, *OV-MDT*, *G-MDT*, *LS-MDT*, *ILS-MDT*, *SA-MDT*, and *RLS-MDT* algorithms.

formances for the whole set of instances because there were no significant differences between these two algorithms (using the Tukey test).

3 Conclusions

In this work four metaheuristic techniques were implemented to find high quality triangulations of minimum dilation. The set of instances generated and used in the experimental evaluation are available at the research project site (www.dirinfo.unsl.edu.ar/bd2/GeometriaComp/)

After performing the experimental evaluation and statistical analysis, we assessed the applicability of the LS, ILS, SA, and RLS algorithms for the MDT problem. The *RLS-MDT* and *ILS-MDT* algorithms had similar competitive performances with respect to the other algorithms in the problem instances considered. These algorithms achieved a dilation reduction between 0.4% and 9.2% with respect to the greedy strategy (*GT-MDT*). Although both algorithms behave similarly, the *RLS-MDT* algorithm required less evaluations than *ILS-MDT*. It should be noted that the existing algorithm (*OV-MDT*) yielded no results for four of the instances considered as either it returned an error or halted. The *RLS-MDT* algorithm outperformed the *OV-MDT* algorithm in 50% of the instances considered and obtained the same results in the other instances.

References

[1] E. Aarts and J. Lenstra. *Local Search in Combinatorial Optimization.* John Wiley, 1997.

[2] T. Bartz-Beielstein. *Experimental Research in Evolutionary Computation: The New Experimentalism (Natural Computing Series).* Springer, 2006.

[3] V. Černý. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 1985.

[4] J. Derrac, S. García, D. Molina, and F. Herrera. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, 1(1):3 – 18, 2011.

[5] J. Edmonds. Matroids and the greedy algorithm. *Mathematical Programming*, 1(1):127–136, 1971.

[6] D. Eppstein. Spanning trees and spanners. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, chapter 9, pages 425–461. Elsevier, 2000.

[7] S. Kirkpatrick, C. Gelatt, and M. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[8] A. Klein. Effiziente berechnung einer dilationsminimalen triangulierung. Master's thesis, 2006.

[9] O. Martin, S. Otto, and E. Felten. Large-step markov chains for the traveling salesman problem. *Complex Systems*, 5:299–326, 1991.

[10] Z. Michalewicz and D. Fogel. *How to Solve It: Modern Heuristics.* Springer, 2004.

[11] C. Papadimitriou. *The complexity of combinatorial optimization problems.* PhD thesis, Princeton, NJ, USA, 1976. AAI7704795.

[12] G. Xia. Improved upper bound on the stretch factor of delaunay triangulations. In *Proceedings of the 27th annual ACM symposium on Computational geometry*, SoCG '11, pages 264–273, New York, NY, USA, 2011. ACM.