Metaheuristic Approaches for MWT and MWPT Problems

Maria Gisela Dorzán[1], Edilma Olinda Gagliardi[1], Gregorio Hernández Peñalver[2], and Mario Guillermo Leguizamón[1]*

[1] Facultad de Ciencias Físico Matemáticas y Naturales
Universidad Nacional de San Luis
San Luis, Argentina
{mgdorzan,oli,legui}@unsl.edu.ar
[2] Facultad de Informática
Universidad Politécnica de Madrid
Madrid, España
{gregorio}@fi.upm.es

Abstract. It is known that the Minimum Weight Triangulation problem is NP-hard. Also the complexity of Minimum Weight Pseudo-Triangulation problem is unknown, suspecting that it is also a NP-hard problem. Therefore we focused on the development of approximate algorithms to find high quality triangulations and pseudo-triangulations of minimum weight. In this work we propose the use of two metaheuristics to solve these problems: Ant Colony Optimization (ACO) and Simulated Annealing (SA). For the experimental study we have created a set of instances for MWT and MWPT problems since no reference to benchmarks for these problems were found in the literature. Through the experimental evaluation, we assess the applicability of the ACO and SA metaheuristics for MWT and MWPT problems. These results are compared with those obtained from the application of deterministic algorithms for the same problems (Delaunay Triangulation for MWT and a Greedy algorithm respectively for MWT and MWPT).

Keywords: Metaheuristics, ACO, SA, Computational Geometry, MWT and MWPT problems

1 Introduction

Special geometric configurations, such as *triangulations* and *pseudo-triangulations*, are interesting to investigate due to their use in many fields of application, e.g. computer graphics, scientific visualization, robotics, computer vision, and image synthesis, rigidity theory and motion planning.

* The authors would like to thank to Research Project Tecnologías Avanzadas de Bases de Datos 22/F014 financed by Universidad Nacional de San Luis, San Luis, Argentina; CONICET-AGENTINA; Instituto de Física Aplicada (INFAP)-UNSL-CONICET to allow us to use the cluster; and Research Project MTM2008-05043 del Ministerio de Ciencia e Innovación-España.

Find globally optimal geometric configurations with respect to the *e ght*, are difficult to be found by deterministic methods, since no polynomial algorithm is known. Indeed, the Minimum Weight Triangulation (MWT) and the Minimum Weight Pseudo-Triangulation (MWPT) problems minimize the sum of the edge lengths, providing a quality measure for determining how good is a structure. Mulzer and Rote [10] recently showed that MWT problem is NP-hard. The complexity of MWPT problem is unknown, suspecting that it is also a NP-hard problem.

The approximate algorithms arise as alternative candidates for these problems. These algorithms can obtain approximate solutions to the optimal ones. They have a simple implementation and they can efficiently find good solutions for NP-hard optimization problems [9].

This paper is organized as follows. In the next Section, we present the theoretical aspects of MWT and MWPT problem. Following, we describe the ACO and SA metaheuristics and the proposed algorithms. Next Section, we present the experimental and statistical study. Last Section is reserved for the conclusions and future vision.

2 Minimum Weight Triangulation and Pseudo-Triangulation problems

Let S be a set of points in the plane. A triangulation of S is a partition of the convex hull of S into triangles whose set of vertices is exactly S. The weight of a triangulation T is the sum of the Euclidean lengths of all the edges of T. The triangulation that minimizes this sum is named a *Minimum Weight Triangulation* of S and it is denoted by $MWT(S)$.

Let S be a set of points in the plane. A pseudo-triangulation PT of S is a partition of the convex hull of S into pseudo-triangles whose set of vertices is exactly S. A pseudo-triangle is a planar polygon that has exactly three convex vertices, called *corners*. The weight of a pseudo-triangulation PT is the sum of the Euclidean lengths of all the edges of PT. The pseudo-triangulation that minimizes this sum is named a *Minimum Weight Pseudo-Triangulation* of S and it is denoted by $MWPT(S)$.

MWT previous results

The MWT problem was first considered by Düppe and Gottschalk [4] who proposed a greedy algorithm which always adds the shortest edge to the triangulation. Later, Shamos and Hoey [15] suggested using the Delaunay triangulation as a minimum weight triangulation. Lloyd [8] provided examples which show that both proposed algorithms usually do not compute the MWT. Many papers present solutions to problems in the field of Graphical Computation. In 1992, Sen and Zheng [14] proposed an algorithm using Simulated Annealing. The neighborhood is obtained with a flip in a random edge of the current triangulation. In 1993, Wu and Wainwright [16] use a genetic algorithm where the recombination and mutation operators are the same (a flip). In the previous mentioned works, the experimental evaluation is rather poor and they do not describe the quality of the obtained solutions. In 2001, Kolingerova and Ferko [7] presented a genetic optimization, which recombination operator is named DeWall and the mutation operator makes a flip in the selected individual. The principal weakness of this method is the time demand. Later, Mulzer and Rote demonstrated in 2006 that MWT's construction is a NP-hard problem [10].

MWPT previous results

The concept of pseudo-triangulation was introduced by Pocchiola and Vegter in [11] on the analogy of the arrangements of pseudo-lines; see [12] for a survey with many results of pseudo-triangulations. There exists a set of points for which any triangulation will have weight $O(n\cdot\ t(M(S)))$. A natural question is whether there exist a similar worst-case bounds for pseudo-triangulations. Rote et al. [13] were those who asked if the MWPT is a NP-hard problem, stimulating the search of exact or approximate algorithms. Gudmundsson and Levcopoulos [5] considered the problem of computing a minimum weight pseudo-triangulation of a set P of n points in the plane, presenting an $O(n\cdot\ ogn)$-time algorithm that produces a pseudo-triangulation of weight $O(\ ogn.\ \ t(M(P)))$ which is shown to be asymptotically worst-case optimal. That is, there exists a point set P for which every pseudo-triangulation has weight $\Omega(\ ogn.\ \ t(M(P)))$, where $\ t(M(P))$ is the weight of a minimum spanning tree of P. Also, they presented a constant factor approximation algorithm running in cubic time, and they gave an algorithm that produces a minimum weight pseudo-triangulation of a simple polygon. It is also worth noticing that to the best knowledge of the authors, there are no results with applying metaheuristics techniques.

3 Ant Colony Optimization Metaheuristic - ACO

The ACO metaheuristic involves a family of algorithms in which a colony of artificial ants cooperate in finding good solutions to difficult discrete optimization problems. An artificial ant in an ACO algorithm is a stochastic constructive procedure that incrementally builds a solution by adding opportunely defined solution components to a partial solution under construction. The ACO metaheuristic can be applied to any combinatorial optimization problem for which a constructive graph can be defined. Each edge (,) in the graph represents a posible path and it has associated two information sources that guide the ant moves: pheromone trails and heuristic information.

Following we present in detail the specific component of the general ACO algorithm (function *BuildSolutionk*) that have to be adapted for MWT and MWPT problems.

Main components of Algorithm 1:

nitiali e: it initializes the parameters:) τ_0 is the initial trail of pheromone is associated to each edge;) K is the quantity of ants of the colony;) α and β are described latter; and v) C is the maximum number of cycles.

BuildSolutionk: this process begins with a partial empty solution which is extended at each step by adding a feasible solution component chosen from the current solution neighbors. The choice of a feasible neighbor is done in a

Algorithm 1 General-ACO

ni ialize
for $\in \{1, \ldots, \}$ **do**
  for $\in \{1, \ldots, \}$ **do**
    B ild ol ion
      al a e ol ion
  end for
  a eBes ol ion o ar
  U da e rails
end for
e rnBes ol ion

probabilistic way in every step of the construction, depending on the used ACO variant. In this work, the selection rule is based on the following probabilistic model:

$$P_{ij} = \begin{cases} \frac{\tau^{\alpha}_{j} . \eta^{\beta}_{j}}{\sum\limits_{h \in F( )} \tau^{\alpha}_{h} . \eta^{\beta}_{h}}, j \in F( ); \\ 0, \qquad\qquad \text{otherwise.} \end{cases} \tag{1}$$

$F( )$ is the set of feasible points for point ; τ_{ij} is the pheromone value associated to edge (,); η_{ij} is the heuristic value associated to edge (,); and, α and β are positives parameters for determining the relative importance of the pheromone with respect to the heuristic information.

UpdateTrails: increases the pheromone level in the promising paths, and is decreased in other case. First, all the pheromone values are decreased by means of the evaporation process. The pheromone level is increased when good solutions appear. The following equation is used:

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \Delta\tau_{ij} \tag{2}$$

- $\rho \in (0, 1]$ is the factor of persistence of the trail.
- $\Delta\tau_{ij} = \sum\limits_{=1}^{K} \Delta \ \tau_{ij}$ is the accumulation of trail, proportional to the quality of the solutions.
- $\Delta \ \tau_{ij} = \begin{cases} Q/L \ , \text{when ant used edge (,);} \\ 0, \quad \text{in other case.} \end{cases}$
- Q is a constant depending of the problem; it usually set to 1.
- L is the objective value of the solution k.

In this work, the update of the pheromone trail can be done according to one of the following criteria: *elitist* and *not elitist*. In the elitist case, the best found solution is used to give an additional reinforcement to the levels of pheromone. The not elitist one uses the solutions found by all the ants to give an additional reinforcement to the levels of pheromone.

The proposed ACO algorithm for MWT (ACO-MWT): *BuildSolutionk* works as follows. Each ant builds a triangulation of a given instance P, starting from an initial random point. At each step, the algorithm adds a new edge (,) if there is no intersection between (,) and the edges of the (partial) solution S . In this case, is a feasible point for and vice versa. If the current point has no feasible points, it selects the next reference point according to one of the following criterions:) random selection;) select the point with the largest quantity of feasible points; or,) select the point with the lowest quantity of feasible points.

The proposed ACO algorithm for MWPT (ACO-MWPT): each ant in *BuildSolutionk* function builds a pseudo-triangulation, starting with one face. This initial face has the edges obtained by the convex hull of the points set P, i.e., $CH(P)$. For the solution construction, each ant performs a process of partitioning the set P in faces. This process finishes when all faces are pseudo-triangles without interior points. A face is divided into two faces when it has interior points or is not a pseudo-triangle. Thus, the partition can be done if) there are at least one interior point and two points in the border; or) there is not an interior point, so it uses two points located on the border.

Parameter settings: we used the following parameter values: $\alpha = 1$; $\beta = 1$ and 5; and $\rho = 0.10$, 0.25, and 0.50. $elit = 1$ and 0, where 1 means that the trail is updated in a elitist way; in other case, the updating is done in a not elitist way. $criterion = 1$, 2, and 3, is used for selecting a point in the *BuildSolutionk* procedure. For $cr\ ter\ on = 1$ the point is chosen randomly; for $cr\ ter\ on = 2$, the chosen point has the largest quantity of feasible points; and for $cr\ ter\ on = 3$, the chosen point has the lowest quantity of feasible points. $C = 1000$. $K = 50$.

4 Simulated Annealing - SA

Simulated Annealing applied to optimization problems emerges from the work of S. Kirkpatrick et al. [6] and V. Čern ́ [3]. SA is based on the principles of statistical mechanics whereby the annealing process requires heating and then slowly cooling a substance to obtain a strong crystalline structure. The SA algorithm simulates the energy changes in a system subjected to a cooling process until it converges to an equilibrium state (steady frozen state). For that, SA introduces a control parameter T, called temperature, to determine the probability of accepting nonimproving solutions (uphill moves). At each iteration, a random neighbor is generated. The moves that improve the cost function are always accepted. Otherwise, the neighbor is selected with a given probability that depends on the current temperature. This probability is usually called *acceptance function* and it is evaluated according to $p(T, \ , \ ) = e^{-\frac{\delta}{T}}$ where $\delta = f(\ ) - f(\ )$.

The basic outline of the General-SA algorithm is illustrated in Algorithm 2.

The algorithm generates an initial solution $\ \in S$, which can be randomly or heuristically constructed, and by initializes the temperature value T_0. Being

Algorithm 2 General-SA

Generate an initial solution $x \in$
Set the initial temperature $_0$
$\leftarrow 0$
while termination condition not met **do**
  $i = 1$
  while $i < \ \ (\ _k)$ **do**
    Generate $y \in \mathcal{N}(x) \subset$
    Evaluate $\delta = \ (y) - \ (x)$
    if $\delta < 0$ **then**
      $x \leftarrow y$
    else
      $x \leftarrow y$ with probability $(\ , x, y)$ // see the *acce ance f nc ion*
    end if
    $i \leftarrow i + 1$
  end while
  $\leftarrow \ + 1$
  Decrease temperature $_k$
end while

$(T$ the number of iterations for temperature T , at each one a new solution $\in \mathcal{N}(\ )$ is randomly generated. If is better than , then is accepted as the current solution. Otherwise (the move from to is an uphill move), is accepted with a probability computed according to the *acceptance function*. Finally, the value of T is decreased at each algorithm iteration . The algorithm continues this way until the termination condition is met. Therefore, considering an optimization problem it is necessary to adapt it to the SA scheme, which is obtained by specifying the following parameters: representation of the solution space S, objective function f, neighborhood of a solution $\mathcal{N}(\ )$, initial solution S_0, initial temperature T_0, temperature decrement rule $\mathcal{R}$, number of moves at each temperature $(T\ )$, and termination condition.

Parameter settings: we describe the parameters for the proposed SA algorithms.

Solution Space S: given a set P of n points in the plane, the solution space for MWT problem is represented by triangulations and for MWPT problem by pseudo-triangulations.

Objective function f: the objective function $f : S \to \mathbb{R}$ assigns to each element of S a real value. For each $S_i \in S$, the function f is defined as the sum of the Euclidean lengths of the edges of the S_i solution.

nitial solution S_0: considering the MWPT and MWT problems, and a set P of n points in the plane, the SA algorithm can start with different initial solutions.

nitial temperature T_0: the initial temperature depends on the number m of edges in the initial solution and the quality measure that is being considered in the problems, i.e., $T_0 = m \times$, where is the average length of the edges of solution S_0.

Temperature decrement rule $\mathcal{R}$: we use three different types of rules: (i) Fast Simulated Annealing ($T_{\ +1} = \frac{T_0}{(1+\ )}$); ($ii$) Very Fast Simulated Annealing ($T_{\ +1} = \frac{T_0}{e^k}$); and ($iii$) Geometric Decrease ($T_{\ +1} = \alpha T$ with $\alpha = 0.8, 0.9$, and 0.95).

Number of moves at each temperature $(T\ )$: we use $(T\ ) = T$ to ensure that the amount of moves is directly proportional to the actual temperature.

Termination condition: the search process is finished when the temperature is less than or equal to 0.005, i.e., $T_f = 0, 005$.

Neighborhood of a solution $\mathcal{N}(\ )$: for each solution $S_i \in S$ it obtains an element $S_j \in S$, called *neighbor*. For MWT and MWPT problems, there are different neighborhood operators.

Neighborhood for MWT: *i) Flip*: an edge is randomly chosen in the current solution and performs the flip operation on , whenever possible. If unable to make the flip, because the edge is illegal, again an edge is chosen at random and repeat this operation.

ii) Local retriangulation: a vertex u of the current solution is randomly chosen and all vertices adjacent to u are recovered. Then the polygon that form the adjacent vertices to u is recovered and the interior of this polygon with the vertex u is retriangulated. This retriangulation can be done in a random or greedy way. In the first, the edges of the polygon are inserted at random until they do not intersect the previously added. In the second way, considering its length, all the edges are sorted in the polygon and are inserted in that order until they do not intersect with the previously added.

Neighborhood for MWPT: Considering an adjacent edge e to two neighboring pseudo-triangle, where each endpoint of e is a corner in at least one of the neighboring pseudo-triangles, since each vertex has at most one convex angle. Removing an edge e merges the two neighboring pseudo-triangles into a *pseudo-quadrangle*. Thus, the six corners of the original two pseudo-triangles become the four corners of a pseudo-quadrangle. A diagonal is defined for a pseudo-quadrangle by connecting opposite corners with a shortest path through the interior. This path coincides with parts of the boundary, except for exactly one straight edge in the interior. There are two diagonals and each one splits the pseudo-quadrangle into two pseudo-triangles. The edge-flip for e removes e and replaces it with the other diagonal [2].

5 Experimental Evaluation and Statistical Analysis

The collections of problem instances were designed by the authors, using an *instance generator* with different functions of CGAL Library [1]. To the best knowledge of the authors there not exist in the literature benchmarking data publicly available that allow us to compare our proposal with other algorithms. A collection of 10 instances of size 40/80/120/160/200 were generated; i.e., a total of 50 problem instances, each one is called LDn- , the size of the -instance,

$1 \leq \quad \leq 10$, is denoted by n. The points are randomly generated, uniformly distributed with coordinates $\quad , \quad \in [0, 1000]$. For implementation purposes, there are non collinear points. The proposed algorithms were implemented in C language and run on BACO parallel cluster under CONDOR batch queuing system.

We analyze the performance of the ACO and SA algorithms for all the combinations over four instances of 40 and 80 points. Then, there are twelve parameter settings for ACO algorithms, and there are five parameter settings for SA algorithms. For each parameter setting, 30 runs were performed using different random seeds.

Analyzing MWT Problem

The best results were obtained with SA-MWT algorithm using local retriangulation neighborhood operator and decrementing the temperature in a geometric way (with $\alpha = 0.95$). Considering the ACO-MWT algorithm and the smallest obtained weights, the four best parameter settings were selected. The configuration $\alpha = 1$, $\beta = 5$, $\rho = 0.50$, and $e \quad t = 1$ is always in the four selected ones. Therefore, we compare statistically both algorithms with such parameter settings.

In Table 1, the best weights and the median values for ACO-MWT and SA-MWT algorithms, the weights for Delaunay Triangulation (DT) and Greedy Triangulation (GT) are showed. As the values do not not follow a normal distribution (using Kolmogorov-Smirnov test) the Kruskal-Wallis test (a nonparametric statistical test) was apply to perform the median comparison in order to determine if there is significant difference between both algorithms (*p-value* column). Generally the values of SA-MWT algorithm are significantly different from the ACO-MWT ones, except for one case. This does not allow us to assert that SA-MWT algorithm is better than ACO-MWT algorithm, but shows a high superiority in performance with respect to the considered instances.

Table 1. Results for MWT problem.

Instance	*O-M* Best	*O-M* Median	*-M* Best	*-M* Median	*- al e*	*D*	*G*
LD40-1	5493047	5502009	5463745	5477181	3.6784E-12	5666348	5477181
LD40-2	4659553	4664817	4659552	4659552	1.11E-12	4722381	4659552
LD40-3	5502567	5519777	5478923	5489487	2.20E-12	5663032	5489487
LD40-4	5745772	5747745	5746236	5751867	0.5142	6289829	5751867
LD80-1	6242505	6273781	6220029	6231682	1.19E-11	6462038	6231682
LD80-2	7603796	7634061	7572419	7581868	1.63E-12	8081573	7581868
LD80-3	5836037	5865538	5828344	5845506	8.02E-11	6143637	5845506
LD80-4	6217040	6283664	6147234	6147234	1.14E-12	6460311	6147234

Analyzing MWPT Problem

Greedy Pseudo-Triangulation (GPT) algorithm builds a pseudo-triangulation starting with one face. This face has the edges obtained by the convex hull of the points set P, i.e., $CH(P)$. SA2P-MWPT algorithm is an improved version of SA-MWPT algorithm, which involves a double pass under certain criteria, considering the best results obtained in some temperatures. SA2P-MWPT algorithm decrementing the temperature in a geometric way (with $\alpha = 0.95$) obtain the best result. With respect to the ACO-MWPT algorithm and the smaller obtained weights, the four best parameter settings were selected. The configuration $\alpha = 1$, $\beta = 5$, $\rho = 0.10$, and $e \quad t = 1$ is always in the four selected ones. Therefore, we compare statistically both algorithms with such parameter settings.

Table 2 shows the results according to the smallest weights obtained using ACO-MWPT, SA-MWPT, SA2P-MWPT and GPT algorithm. We performed the Kolmogorov-Smirnov test to show the sample does not follow a normal distribution. Therefore we use a non-parametric statistical test to evaluate the algorithms. Kruskal-Wallis test was performed to compare the medians for determining if there is significant difference between both algorithms (*p-value* column). SA2P-MWPT algorithm obtained the best results with respect to others algorithms. Considering ACO-MWPT and SA2P-MWPT algorithms the Kruskal-Wallis test determined similar performances.

Table 2. Results for MWPT problem.

Instance	*O-* MWPT Best	*O-* MWPT Median	- MWPT Best	- MWPT Median	*P-* MWPT Best	*P-* MWPT Median	*- al e*	*GP*
LD40-1	6115636	6607908	6252359	6907139	4709719	5764730	3.65E-7	5312131
LD40-2	4442710	4757694	5197488	5602824	4812286	5114311	3.17E-11	4292347
LD40-3	5684342	6071705	6017744	7209617	4774148	6182200	0.0625	5794018
LD40-4	5627098	6258985	6133612	6883127	4273332	5856490	0.0003	6245196
LD80-1	7898497	8300956	8428879	10379962	6090317	8740421	0.1008	7458787
LD80-2	9584718	10604489	10197976	12037685	8265211	10206975	0.4965	8931272
LD80-3	8918853	9565227	8265748	10260313	6484145	8365633	5.10E-5	6516103
LD80-4	8004652	9565227	8768465	10473020	6238395	7840248	0.0891	7393297

6 Conclusions

Minimum Weight Triangulation problem is NP-hard and the complexity of Minimum Weight Pseudo-Triangulation problem is unknown, suspecting is NP-hard problem. We proposed approximate algorithms to find high quality geometric configurations of minimum weight.

In this work, we development the use of two metaheuristics to solve these problems: Ant Colony Optimization (ACO) and Simulated Annealing (SA), and deterministic techniques (Delaunay Triangulation for MWT and a Greedy algorithm for MWT and MWPT)respectively.

The set of instances was created since no reference to benchmarks for these problems were found in the literature.

Considering the experimental evaluation, we assess the applicability of the ACO and SA metaheuristics for MWT and MWPT problems. Theses performances were analyzed nonparametric statistical tests.

Also, these results were compared with those obtained from the application of deterministic algorithms for the same problems (Delaunay Triangulation for MWT and a Greedy algorithm respectively for MWT and MWPT).

The statistical analysis showed SA-MWT algorithm has a high superiority in performance; and for ACO-MWPT and SA2P-MWPT algorithms have similar performances, although SA2P-MWPT algorithm obtained the best results with respect to others algorithms.

References

1. Cgal, computational geometry algorithms library, `http://www.cgal.org`
2. Brönnimann, H., Kettner, L., Pocchiola, M., Snoeyink, J.: Counting and enumerating pointed pseudotriangulations with the greedy flip algorithm. SIAM J. Comput. 36, 721–739 (September 2006), `http://portal.acm.org/citation.cfm?id=1237637.1237645`
3. ˇerný, V.: Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. Journal of Optimization Theory and Applications (1985)
4. Düppe, R., Gottschalk, H.: Automatische interpolation von isolinien bei willkürlichen stützpunkten. Allgemeine Vermessungsnachrichten 77, 423–426 (1970)
5. Gudmundsson, J., Levcopoulos, C.: Minimum weight pseudo-triangulations. Comput. Geom. 38(3), 139–153 (2007)
6. Kirkpatrick, S., Gelatt, C., Vecchi, M.: Optimization by simulated annealing. Science 220, 671–680 (1983)
7. Kolingerová, I., Ferko, A.: Multicriteria-optimized triangulations. The Visual Computer 17(6), 380–395 (2001)
8. Lloyd, E.: On triangulations of a set of points in the plane. In: "Proc. 18th IEEE Symp. Found. Comp. Sci. pp. 228–240 (1977)
9. Michalewicz, Z., Fogel, D.: How to Solve It: Modern Heuristics. Springer (2004)
10. Mulzer, W., Rote, G.: Minimum weight triangulation is np-hard. In: In Proc. 22nd Annu. ACM Sympos. Comput. Geom. pp. 1–10. ACM Press (2006)
11. Pocchiola, M., Vegter, G.: Pseudo-triangulations: Theory and applications. In: Symposium on Computational Geometry. pp. 291–300 (1996)
12. Rote, G., Santos, F., Streinu, I.: Pseudo-triangulations — a survey. Contemporary Mathematics, American Mathematical Society (December 2008)
13. Rote, G., Wang, C., Wang, L., Xu, Y.: On constrained minimum pseudotriangulations. In Proc. 9th Intern. Comp. Comb. Conf. (2003)
14. Sen, S., Zheng, S.: Near-optimal triangulation of a point set by simulated annealing. In: SAC '92: Proceedings of the 1992 ACM/SIGAPP symposium on Applied computing. pp. 1000–1008. ACM, New York, NY, USA (1992)
15. Shamos, M., Hoey, D.: Closest-point problems. In: Proc. 16th IEEE Symp. Foundations of Comp. Science. pp. 151–162 (1975)
16. Wu, Y., Wainwright, R.: Near-optimal triangulation of a point set using genetic algorithms. In: Proceedings of the Seventh Oklahoma Conference on AI. (1993)