Approximations on Minimum Weight Pseudo-Triangulations using Ant Colony Optimization Metaheuristic

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

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

Abstract. Globally optimal pseudo-triangulations are difficult to be found by deterministic methods as, for most type of criteria, no polynomial algorithm is known. In this work, we consider the Minimum Weight Pseudo-Triangulation (MWPT) problem of a given set of n points in the plane. This paper shows how the Ant Colony Optimization (ACO) metaheuristic can be used to find optimal pseudo-triangulations of minimum weight. For the experimental study presented here we have created a set of instances for MWPT since no reference to benchmarks for these problems were found in the literature. We assess through the experimental evaluation the applicability of the ACO metaheuristic for MWPT.

Key words: Pseudo-Triangulation, Minimum Weight, Computational Geometry, ACO Metaheuristic

1 Introduction

In Computational Geometry there are many problems that either are NP-hard or no polynomial algorithms are known. Therefore, it is interesting to find approximate solutions using metaheuristics. The optimization problems related to special geometric configurations, such as *triangulations* and *pseudo-triangulations*, are interesting to research due to their use in many fields of application, e.g., visibility, ray-shooting, kinetic collision detection, rigidity, guarding, etc. The pseudo-triangulations, like triangulations, are planar partitions. Minimizing the total length has been one of the main optimality criteria. The related problems are the Minimum Weight Triangulation (MWT) and the Minimum Weight Pseudo-Triangulation (MWPT), that minimize the sum of the edge lengths, providing a quality measure for determining how good a structure is. The complexity of computing a minimum weight triangulation has been one of the most long-standing open problems in Computational Geometry, introduced by Garey and Johnson [5] in their open problems list, and various approximation algorithms were proposed over time. Mulzer and Rote [7] recently showed that MWT problem is NP-hard. The complexity of MWPT problem is unknown, but Levcopoulos and Gudmundsson [6] show that a 12-approximation of an MWPT can be computed in $O(n^3)$ time. They give an $O(\log n \cdot w(MST))$ approximation of an MWPT, in $O(n \log n)$ time, where $w(MST)$ is the weight of the minimum Euclidean spanning tree, which is a subset of the obtained structure.

Given the inherent difficulty of these problems, the approximate algorithms arise as alternative candidates. These algorithms can obtain approximate solutions to the optimal solutions, and they can be specific for a particular problem or they can be part of a general applicable strategy in the resolution of different problems. The metaheuristic methods satisfy these properties.

In this work, we consider the MWPT problem of a given set of n points in the plane. This paper shows how the Ant Colony Optimization (ACO) metaheuristic can be used to find optimal pseudo-triangulations of minimum weight. For the experimental study presented in this work we use the *Ant Colony Optimization* (ACO) metaheuristic.

This paper is organized as follows. In the next section we present the theoretical aspects of pseudo-triangulations. In Section 3, we present the general overview of the ACO metaheuristic. Section 4 presents the proposed ACO algorithms for the MWPT problem. Finally, we describe the MWPT instances used and the details and results of the experimental study in which we analyze the sensitivity of some important parameters on the performance of the proposed ACO algorithm.

2 Minimum Weight Pseudo-Triangulation

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 vertex set 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 $MWPT(S)$.

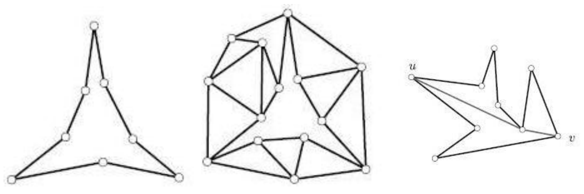


Fig. 1. (a) A pseudo-triangle, (b) a pseudo-triangulation of a point set and (c) a pseudo-triangulation of a simple polygon, including a geodesic path from u to v [9].

The concept of pseudo-triangulation was introduced by Pocchiola and Vegter in [8] on the analogy of the arrangements of pseudo-lines; see [9] for a survey with many results of pseudo-triangulations. As we mentioned in Section 1, there exists a set of points for which any triangulation will have weight $O(nwt(M(S)))$. A natural question is whether there exist a similar worst-case bounds for pseudo-triangulations. Rote et al.,[10] were those who asked if the MWPT is a NP-hard problem, stimulating the search of exact or approximate algorithms. Gudmundsson and Levcopoulos [6] considered the problem of computing a minimum weight pseudo-triangulation of a set S of n points in the plane, presenting an $O(nlogn)$-time algorithm that produces a pseudo-triangulation of weight $O(logn.wt(M(S)))$ which is shown to be asymptotically worst-case optimal. That is, there exists a point set S for which every pseudo-triangulation has weight $\Omega(logn.wt(M(S)))$, where $wt(M(S))$ is the weight of a minimum spanning tree of S. 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. Previous works about approximations on mentioned problems using metaheuristic, were presented in [2] and [3] , where we described the design of the ACO algorithms and gave the first steps in this research.

3 Ant Colony Optimization Metaheuristic

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. Cooperation is a key design component of ACO algorithms: The choice is to allocate the computational resources to a set of relatively simple agents (artificial ants) that communicate indirectly by stigmergy. Thus, good quality solutions are an emergent property of the agents cooperative interaction.

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. Therefore, the ACO metaheuristic can be applied to any combinatorial optimization problem for which a constructive graph can be defined. Each edge (i, j) in the graph represents a posible path and it has associated two information sources that guide the ant moves: pheromone trails and heuristic information. The pheromone trail, denoted by τ_{ij}, encodes a long-term memory about the entire ant search process, and is updated by the ants themselves. The heuristic information, denoted by η_{ij}, represents *a priori* information about the problem instance or run-time information provided by a source different from the ants. In many cases η is the cost, or an estimate of the cost, of adding the component or connection to the solution under construction. These values are used by the ants to make probabilistic decisions on how to move on the graph. The ants act concurrently and independently and although each ant is complex enough to find a solution to the problem, which is probably poor, good-quality solutions can only emerge as the result of the collective interaction among the ants. This is obtained via indirect communication mediated by the information ants read or write in the variables storing pheromone trail values. It is a distributed learning process in which the single agents, the ants, are not adaptive themselves but, on the contrary, adaptively modify the way the problem is represented and perceived by other ants [4].

There are two additional process for updating pheromone and the daemon actions. The pheromone updating is the process by which the pheromone trails are modified. The trail values can either increase, as ants deposit pheromone on the components or connections they use, or decrease, due to pheromone evaporation. The daemon procedure is used to implement centralized actions which cannot be performed by single ants. Examples of daemon actions are the activation of a local optimization procedure, or the collection of global information that can be used to decide whether it is useful or not to deposit additional pheromone to bias the search process from a nonlocal perspective. The daemon can observe the path found by each ant in the colony and select one or a few ants, like those that built the best solutions in the algorithm iteration that allowed to deposit additional pheromone on the connections they used.

3.1 The general ACO algorithm

In this section we present a general ACO algorithm (Algorithm 1) and a description of the main components. After that, the next section describes in detail the specific component of the general ACO algorithm (function *BuildSolutionk*) that have to be adapted for MWPT. Main components of Algorithm 1:

Algorithm 1 General-ACO

```
Initialize
for $c \in \{1, \dots, C\}$ do
    for $k \in \{1, \dots, K\}$ do
        BuildSolutionk
        EvaluateSolution
    end for
    SaveBestSolutionSoFar
    UpdateTrails
end for
ReturnBestSolution
```

Initialize: this process initializes the parameters considered for the algorithm. The initial trail of pheromone is associated to each edge, τ_0; it is an small positive value, in general, the same for all edges. The quantity of ants of the colony, K. The weights that define the proportion in which they will affect the heuristic information and pheromone trails in the probabilistic transition rule, named respectively β y α. 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; i.e., to find a route on the construction graph guided by the mechanism that defines the set of feasible neighbors with regard to the partial solution. The choice of a feasible neighbor is done in a probabilistic way in every step of the construction, depending on the used ACO variant. In this work, the selection rule for the solutions construction is based on the following probabilistic model:

$$P_{ij} = \begin{cases} \frac{\tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}}{\sum\limits_{h \in F(i)} \tau_{ih}^{\alpha} \cdot \eta_{ih}^{\beta}}, & j \in F(i); \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

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

EvaluateSolution: evaluates and saves the best solution found by ant k in the current cycle.

SaveBestSolutionSoFar: saves the best solution found for all cycles so far.

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 process of evaporation. Then, 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.
- $\Delta\tau_{ij} = \sum\limits_{k=1}^{K} \Delta^k\tau_{ij}$ is the accumulation of trail, proportional to the quality of the solutions.
- $\Delta^k\tau_{ij} = \begin{cases} Q/L_k, & \text{when ant } k \text{ used edge } (i, j); \\ 0, & \text{in other case.} \end{cases}$
- Q is a constant depending of the problem; usually is set to 1.
- L_k is the objective value of the solution k.

Pheromone evaporation avoids a fast convergence of the algorithm. In addition, this way of forgetting allows the exploration of new areas of the search space. The update of the pheromone trail can be done according to one of the following criteria: *elitist* and *non-elitist*. In the elitist case, the best found solution is used to give an additional reinforcement to the levels of pheromone. The non-elitist one uses the solutions found by all the ants to give an additional reinforcement to the levels of pheromone.

3.2 The proposed ACO algorithm for MWPT (ACO-MWPT)

For ACO-MWPT, each ant in *BuildSolutionk* function builds a pseudo-triangulation, starting with one face. This face has the edges obtained by the convex hull of the points set S, i.e., $CH(S)$. For the solution construction, each ant performs a process of partitioning $CH(S)$ 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 *i*) there are at least one interior point and two points in the border; or *ii*) there is not an interior point, so the procedure use two points located on the border. $Faces_k$ represents the set of no treated faces.

Algorithm 2 BuildSolutionk

```
$S_k \leftarrow \emptyset$   /* solution builded by $k$-ant */
while ($Faces_k \neq \emptyset$) do
    Let $F$ be a face in $Faces_k$
    if $F$ is Pseudo-triangle without interior points then
        $S_k \leftarrow S_k \cup \{F\}$  /* $F$ is a new pseudo-triangle */
        $Faces_k \leftarrow Faces_k - \{F\}$
    else
        $PartitionFace(F)$
    end if
end while
```

PartitionFace(F) selects the points of F to build the new faces. It takes one interior point and two probabilistic selected points of the border, or (if there is not interior point) only two probabilistic selected points of the border. The feasible points for a point are those visible and not adjacent to it. The selection is done according to Eq. 1. Also, this process uses two additional procedures *SelectInteriorPoint*(F) and *SelectBorderPoint*(F), where the point selection is achieved according to one of the following criteria: *i*) at random; *ii*) the largest quantity of feasible points; or, *iii*) the lowest quantity of feasible points.

4 Experimental Evaluation

For the whole experimental evaluation, we developed a *generator of points*, using different functions of random generation of CGAL Library [1]. The points in the plane are non collinear, uniform distributed, with coordinates in (0, 1000). The set of instances has been formed by 5 collections of 10 points sets of different cardinality: 40/80/120/160/200-collection respectively. We have the 40-collection with the point sets LD401, LD402,.., LD410; and so on. We emphasize that these collections are not available in previous related researches, and contribute in this paper. The ACO-MWPT algorithms has been implemented in C programming language.

We show the initial experimental phase, that will allow us to decide which are the most suitable parameter settings. The analyzed collections correspond to LD401, LD402, LD403 and LD404. The parameter setting, are according to the Eq. 1 and 2, corresponding to (α-β-ρ-*elit*-*criterion*), where α: 1; β: 1 and 5; ρ: 0.1, 0.25 and 0.5; . If *elit* is set to 1, the trail is update in a elitist way; in other case, the updating is done in a not elitist way. The *criterion* and τ are set to 1. So, there are twelve parameter setting, and for each were performed 30 runs by using different random seeds. The number of cycles, C, is 1000; the number of ants, K, is 50. We obtain average, median, best, and variance/standard deviation values, considering the objective function (*weight*); and quantity of pseudo-triangles. Then, we select only the four best parameter settings. The numerical work were done using the *BACO* parallel cluster, composed by 60 PCs, with a 3.0 GHz Pentium-4 processor each one and 90 PCs with a 2.4 GHz Core 2 Quad processor each one, under *CONDOR* batch queuing system.

Next, the results for this experimental study are resumed in Tables 1 to 6. In Tables 1 to 4, we show the results according to the four best parameter settings with respect to the smaller weights. The configuration (1-5-25-1) obtained the smallest weight for the LD402. See Table 2. Table 5 shows a comparison between the four best parameter settings for the obtained smaller weight values and the four best parameter settings for the obtained smaller average values. The values are similar. Better results were obtained by β: 5; *elit*: 1; ρ: 0.1, 0.25 and 0.5. We obtained better results giving more relevance to the heuristic information and updating the trails in a elitist way. We considerer necessary to observe the influence of parameters in general. Table 6 shows the influence percentage for each parameter with respect to the best found weights values. Likewise, we observe what happened with the best average values. Additionally, in Table 7, we show the influence percentage of every parameter for the best performance with respect to the average. We can observe that the percentages are similar. Figure 2 shows the boxplots of the weights obtained, for the 30 seeds for LD401, LD402 and LD403 for the four best configurations.

Figure 2, from a) to d), show the boxplots of the weights obtained for the 30 seeds for LD401, LD402, LD403 and LD404 for the four best configurations. The boxplots are not similar; for the LD402, the best weights obtained are outliers and has a good behavior. In the case of LD404, the boxplots are regular, but the best values are not approximate to the minimum.

In this sense, Tables 6 and 7 give a clearer idea over which are the suitable parameter settings.

With respect to the quantity of pseudo-triangles found in the most approximate pseudo-triangulation we can observe that not necessarily it is minor.

Table 1. MWPT: Results for LD401.

Par. Setting	Average	Median	Best	Std. Dev.	#PT
1-5-10-1	6557443,73	6607908,75	6115636,63	166770,70	51
1-5-25-1	6644026,13	6658654,75	6286985,04	166104,26	48
1-5-50-1	6669542,40	6713656,75	6320652,56	159069,87	49
1-5-50-0	6777518,93	6847951,25	6322956,39	158225,32	52

Table 2. MWPT: Results for LD402.

Par. Setting	Average	Median	Best	Std. Dev.	#PT
1-5-25-1	4748353,60	4757694,50	4442710,43	114885,64	49
1-5-10-0	4681136,53	4685804,75	4470550,00	69468,03	48
1-5-10-1	4707699,73	4747199,25	4490214,33	83905,98	50
1-5-25-0	4729018,67	4749318,25	4524206,34	77542,87	43

Table 3. MWPT: Results for LD403.

Par. Setting	Average	Median	Best	Std. Dev.	#PT
1-5-25-1	6069210,67	6071705,00	5684342,58	143063,20	49
1-5-10-1	5980440,00	6021063,00	5699513,63	136174,19	50
1-5-25-0	6075029,87	6118394,25	5744775,81	110439,30	45
1-5-50-0	6073308,80	6104511,25	5746463,24	121285,65	51

Table 4. MWPT: Results for LD404.

Par. Setting	Average	Median	Best	Std. Dev.	#PT
1-1-50-1	6236883,73	6258985,50	5627098,22	218860,42	48
1-5-10-1	6162888,00	6154961,25	5668910,18	166455,96	49
1-5-50-1	6229822,93	6237045,50	5869145,72	202030,98	50
1-5-50-0	6237883,20	6252135,50	5903381,03	139767,35	47

Table 5. MWPT: Comparison between four best BEST and AVERAGE parameter settings.

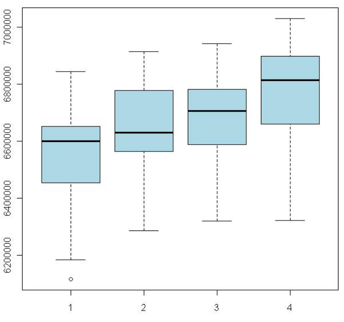
Par. Setting	Best	Par. Setting	Average
LD402 1-5-25-1	4442710,43	LD402 1-5-10-0	4681136,53
LD404 1-1-50-1	5627098,22	LD403 1-5-10-1	5980440,00
LD403 1-5-25-1	5684342,58	LD404 1-5-10-1	6162888,00
LD401 1-5-10-1	6115636,63	LD401 1-5-10-1	6557443,73

Table 6. MWPT: Abstract Table for LD401, LD402, LD403, and LD404 w.r.t. best BEST values.

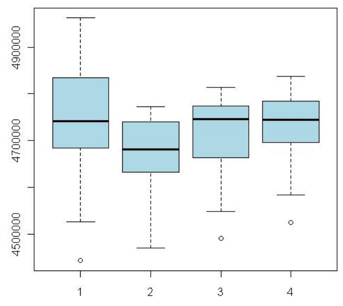
α	β	ρ	*elit*
1 (100%)	5 (95%)	0.1 (31.25%)	1 (62.50%)
	1 (5%)	0.25 (31.25%)	0 (37.50%)
		0.5 (37.50%)	

Table 7. MWPT: Abstract Table for LD401, LD402, LD403, and LD404 w.r.t. best AVERAGE values.

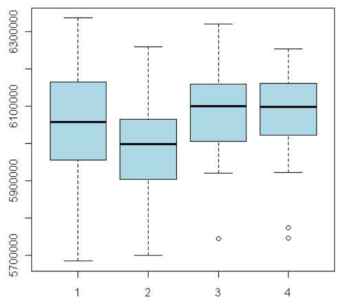
α	β	ρ	*elit*
1 (100%)	5 (88%)	0.1 (37.50%)	1 (62.50%)
	1 (12%)	0.25 (25.00%)	0 (37.50%)
		0.5 (37.50%)	



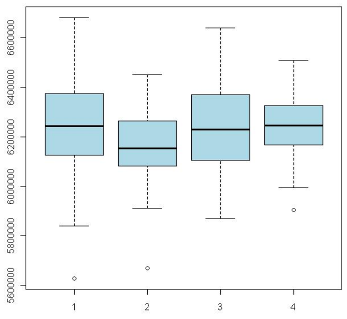
(a)



(b)



(c)



(d)

Fig. 2. MWPT: Boxplots for a) LD401, b) LD402, c) LD403 and d) LD404.

5 Conclusion

In this work, we present the design of approximation algorithms for solving the Minimum Weight Pseudo-Triangulation problems for sets of points in the plane. The proposed ACO model for the MWPT is represented by an Ant System (AS), a particular instance of the class of ACO algorithms. We also detailed the generation of sets of points for the experimental evaluation, considering that is an other contribution. We show the initial experimental phase, where we have preliminary results that guide the future experimentation. So, from our analysis of these approximation algorithms, we obtained the suitable parameter setting. The future work will consist to continue the experimentation with the total collection of set of points and to compare the proposed algorithms against other strategies like Greedy or others metaheuristics. Finally, since the metaheuristics have proven to behave very well in solving this class of NP-hard problem, there are several directions for further research.

Acknowledgment

The authors would like to thank to Research Project Tecnologías Avanzadas de Bases de Datos 22/F614 financed by Universidad Nacional de San Luis, San Luis, Argentina; to the Project AL09-PAC-12 financed by Universidad Politécnica de Madrid, Madrid, España; and, to the Instituto de Física Aplicada, Universidad Nacional de San Luis - CONICET, San Luis, Argentina, to allow us to use the cluster. Likewise, to the colleagues who contributed with opinions and technical advice.

References

1. Computational Geometry Algorithms Library (CGAL). http://www.cgal.org/
2. M. Dorzán, E. Gagliardi, M. Leguizamón and G Hernández Peñalver, *Algoritmo ACO aplicado a la obtención aproximada de Triangulaciones de Peso Mínimo.* XXXV Conferencia Latinoamericana de Informática (CLEI), 2009.
3. M. Dorzán, E. Gagliardi, M. Leguizamón, M. Taranilla and G. Hernández Peñalver, *Algoritmos ACO aplicados a problemas geométricos de optimización.* XIII Encuentros de Geometría Computacional (EGC09), 2009.
4. M. Dorigo and T. Stützle, *Ant Colony Optimization.* Massachusetts Institute of Technology, 2004.
5. M. Garey and D. Johnson, *Computers and Intractability.* W. H. Freeman and Company, 1979.
6. J. Gudmundsson and C. Levcopoulos, *Minimum weight pseudo-triangulations.* Computational Geometry, Theory and applications, 38, pp. 139-153, 2007.
7. W. Mulzer and G. Rote, *Minimum weight triangulation is NP-hard.* Proceedings of the 22nd Annual ACM Symp. on Computational Geometry. pp. 1-10, 2006.
8. M. Pocchiola and G. Vegter, *Pseudo-triangulations: theory and applications.* Proceedings of the 12th Annual ACM Symposium on Computational Geometry, pp. 291-300, 1996.
9. G. Rote, F. Santos and I. Streinu, *Pseudo-triangulations - a survey.* Surveys on Discrete and Computational Geometry-Twenty Years Later, Contemporary Mathematics, E. Goodman, J. Pach, R. Pollack, eds. American Mathematical Society, 2008.
10. G. Rote, C. Wang, L. Wang and Yinfeng Xu, *On constrained minimum pseudo-triangulations.* Computing and Combinatorics: 445-454, Springer-Verlag, Lecture Notes in Computer Science 2697, 2003.