Simulated Annealing applied to the MWPT problem

E.O. Gagliardi*[1], M. G. Leguizamón †[1], and G. Hernández ‡[2]

[1]Universidad Nacional de San Luis, Argentina
[2]Universidad Politécnica de Madrid, España

Abstract

The Minimum Weight Pseudo-Triangulation (MWPT) problem is suspected to be NP-hard. We show here how Simulated Annealing (SA) can be applied for obtaining approximate solutions to the optimal ones. To do that, we applied two SA algorithms, the basic version and our extended hybrid version of SA. Through the experimental evaluation and statistical study we assess the applicability and performance of the SA algorithms. The obtained results show the benefits of using the hybrid version of SA to achieve improved and higher quality solutions for the MWPT problem.

Introduction

A pseudo-triangle is a simple polygon with three convex vertices, and a pseudo-triangulation is a partition of a planar region into pseudo-triangles. See [9] and [10] for surveys about theory and applications, with several interesting results that include combinatorial properties and counting of special classes, rigidity theoretical results, and representations as polytopes, among others.

Optimization problems related to pseudo-triangulations are interesting under several optimality criteria. In this work, we consider the optimality criterion for pseudo-triangulations refereed as Minimum Weight. The weight of a pseudo-triangulation is the total length of their edges. Minimizing the total length is one of the main optimality criteria that provides a quality measure. This formulation is known as the Minimum Weight Pseudo-Triangulation (MWPT) problem. The complexity of the MWPT problem is unknown an it is assumed to be in NP-hard class [6].

Indeed, since no polynomial algorithm is known, approximate solutions of high quality are difficult to obtain by deterministic methods. Thus, we consider approximation algorithms. These algorithms are capable of obtaining approximate solutions to the optimal ones and they can be easily implemented for finding good solutions in NP-hard optimization problems [8].

In this work, we propose the use of *Simulated Annealing* (SA) for finding high quality pseudo-triangulations of minimum weight. For this, we investigate its application through an experimental study and an extended statistical analysis of the results. We have implemented the algorithms involved and additionally, generated our set of instances. Non parametric statistical tests were applied for assessing the performance of the algorithms implemented.

Our recent work on this research, [4] and [5], summarize successive stages of this research using a different metaheuristic. The preliminary results obtained at the initial phase guided to apply a more methodological approach in this research.

This paper is organized as follows. Section 1 describes a general overview of the SA metaheuristic and the main algorithms. Section 2 describes the experimental design, and Section 3 presents the experimental evaluation and statistical analysis. Lastly, Section 4 is devoted to the conclusions.

1 Simulated Annealing

Simulated Annealing applied to optimization problems emerges from the work of S. Kirkpatrick et al. [7] and V. Černý [2]. 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. This case is based in an extension of local search (a trajectory-based approach) to solve combinatorial optimization problems. Without loss of generality, the strategy is good for optimization problems, since SA has an explicit strategy to escape from locally optimal solutions. SA introduces a control parameter, T, called *temperature* or *cooling schedule*, whose initial value should be high and should decrease during the search process. The search process is done according to the

*Email: oli@unsl.edu.ar Research supported by Project Tecnologías Avanzadas de Bases de Datos (22/F014), Universidad Nacional de San Luis, Argentina

†Email: legui@unsl.edu.ar - Research supported by LIDIC, Universidad Nacional de San Luis, Argentina

‡Email: gregorio@fi.upm.es - Research supported by ESF EUROCORES programme EuroGICA. ComPoSe IP04-MINCINN Project EUI-EURC-2011-4306

execution of several iterations of the algorithm until a termination condition is achieved.

In order to apply SA to the MWPT problem it is necessary to define some components of a SA by specifying the following parameters: solution space S, objective function f, neighborhood of a solution $\mathcal{N}(S_i)$, initial solution S_0, initial temperature T_0, temperature decrement rule $\mathcal{R}$, number of moves at each temperature $M(T_k)$ (*length of the Markov chain*), acceptance function, and termination condition. The objective of this study is assessing throughout a rigorous experimental study the applicability and respective performances of MWPT-SA and MWPT-SA-2P for the MWPT problem. Next, we describe the common components taken into account in the experimentation.

1.1 The MWPT-SA Algorithm

This algorithm is the result of two combined strategies: random walk and iterative improvement, commonly named diversification and intensification. The search has two phases. The first phase consists of the exploration of the search space; however, this behavior is slowly decreased, leading the search to converge to a local minimum, i.e., phase of iterative improvement. At each iteration, a neighbor of the neighborhood is randomly chosen. The neighborhood of a pseudo-triangulation is obtained by application of edge flips on two adjacent pseudo-triangles [1]. 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. The basic outline is illustrated in Algorithm 1 (MWPT-SA).

Algorithm 1 MWPT-SA

```
Generate an initial solution S_i ∈ S
Set the initial temperature to T_0
k ← 0
while termination condition not met do
   c ← 1
   while c < M(T_k) do
      Choose S_j ∈ N(S_i) ⊂ S
      Evaluate δ = f(S_j) − f(S_i)
      if δ < 0 then
         S_i ← S_j
         SaveBestSoFarSolution
      else
         S_i ← S_j with probability p(T_k, S_i, S_j)
      end if
      c ← c + 1
   end while
   k ← k + 1
   Decrease temperature T_k
end while
```

Algorithm 1 (MWPT-SA) starts with an initial solution $S_i \in S$, which can be randomly or heuristically constructed. Then, it initializes the temperature with value T_0. $M(T_k)$ is the number of iterations for temperature T_k. At each inner iteration, a new solution $S_j \in \mathcal{N}(S_i)$ is randomly generated. If S_j is better than S_i, then S_j is accepted as the current solution. Otherwise, the move from S_i to S_j is an uphill move, and S_j is accepted with a probability computed according to the acceptance function. Finally, the value of T_k is decreased at each outer iteration, controlled by variable k. The algorithm continues in this way until the termination condition is met.

1.2 The MWPT-SA-2P Algorithm

This is an extend scheme, called MWPT-SA-2P algorithm, which was designed considering the cooling schedule of MWPT-SA. The cooling schedule is used for balancing between diversification and intensification, by allowing to return to previous stages. In this manner, the performance of MWPT-SA is improved. Indeed, escaping from the area of low quality in the phase of diversification was sometimes almost impossible for the basic algorithm. Then it is necessary to have the possibility of exploring other regions of the search space. The basic outline is illustrated in Algorithm 2. The algorithm introduces an additional variable (named *previous*) in order to incorporate a control over the trajectory. This variable allows to know which is the parameter setting of the algorithm running where the best so far solution has been found.

Algorithm 2 MWPT-SA-2P

```
Generate an initial solution S_i ∈ S
Set the initial temperature to T_0
k ← 0
while termination condition not met do
   c ← 1
   while c < M(T_k) do
      Choose S_j ∈ N(S_i) ⊂ S
      Evaluate δ = f(S_j) − f(S_i)
      if δ < 0 then
         S_i ← S_j
         SaveBestSoFarSolution
         T_previuos ← T_k
      else
         S_i ← S_j with probability p(T_k, S_i, S_j)
      end if
      c ← c + 1
   end while
   k ← k + 1
   if the best so far solution was updated then
      Decrease temperature T_k
   else
      if it is the first pass on T_k then
         Return to previous temperature T_previuos and
         Do the second pass
      end if
   end if
end while
```

Algorithm 2 (MWPT-SA-2P) controls, before decreasing the temperature, whether during the current temperature T_k the algorithm has found a better solution than the best so far solution. If not, the process returns to a state refereed as the previous temperature, named $T_{previuos}$, where the best so far solution was found. From that state, the algorithm chooses moves for walking in other directions for exploring other areas of the unexplored solution space. In this case, the SaveBestSoFarSolution process saves the best solution found for all cycles so far and the corresponding temperature $T_{previuos}$. The amount of repetitions to get through T_k was experimented, and we conclude that a maximum of two is enough. As the title suggests, 2P in MWPT-SA-2P stands for *double pass*, as the algorithm returns at most once on the path traveled.

2 Experimental Design

Representation. The pseudo-triangulations are planar subdivisions induced by planar embeddings of graphs. For their representation we use a *Doubly-Connected Edge List* (DCEL) [3]. For evaluation purposes in SA, a solution must be transformed from the DCEL into a nxn matrix, where n is the number of points.

Objective Function. The weight of a pseudo-triangulation PT, named $f_w(PT)$, is the sum of the euclidean lengths of all the edges of PT.

Instances collection. An ad-hoc software was designed and implemented by the authors for generating the collection of problem instances, each one being a set P of n points in the plane. A collection of ten (10) instances of size n were generated, with n equal to 40/80/120. Each one is called LDni, $1 \leq i \leq 10$. The points were randomly generated, uniformly distributed, with coordinates $x, y \in [0, 1000]$. For implementation purposes, there are non collinear points.

Parameter Settings for the SA algorithms. The proposed algorithms were executed thirty (30) times using different random seeds for the complete collection of instances. The initial solution is a pseudo-triangulation. We consider two types of initial solutions for MWPT-SA-2P; the initial pseudo-triangulation is: a) a randomly generated solution and b) a pseudo-triangulation obtained by the GPT algorithm [4]. The initial temperature T_0 depends on the number m of edges in the initial solution and the objective function f_w. $T_0 = m \times l$, where l is the average length of the edges of the initial solution. The number of moves at each temperature $N(T_k)$ is $N(T_k) = T_k$ ensuring that the amount of moves is directly proportional to the actual temperature. In each case, for the Temperature decrement rule $\mathcal{R}$, three different types of rules were considered: (*i*) Fast Simulated Annealing ($T_{k+1} = \frac{T_0}{(1+k)}$); (*ii*) Very Fast Simulated Annealing ($T_{k+1} = \frac{T_0}{e^k}$); and (*iii*) Geometric Decrease ($T_{k+1} = \alpha T_k$, where $\alpha \in [0, 1]$). The Geometric cooling scheme had the best performance according to previous experiments, therefore it was chosen for the study presented in this work. For this cooling scheme, we consider $\alpha = 0.8, 0.9$, and 0.95. The setting $\alpha = 0.95$ was chosen due to its high performance. For the termination condition, the search process is finished when the temperature is less than or equal to $T_f = 0.005$.

Resources. The algorithms were implemented in C and, for the statistical analysis, $MATLAB$ was used.

3 Experimental Evaluation and Statistical Analysis for the proposed SA algorithms

This section shows the applicability of MWPT-SA and MWPT-SA-2P algorithms through experimental evaluation. The initial solution can be a randomly solution generated, or a greedy pseudo-triangulation obtained by the GPT algorithm. SA using the last strategy for generating the initial solution can be considered a hybrid approach. We reference the MWPT-SA-2P algorithm as MWPT-SA-2P-RPT (RPT stands for Random Pseudo-Triangulation) or MWPT-SA-2P-GPT (GPT stands for Greedy Pseudo-Triangulation). The experimental and statistical study considers the mentioned set of instances, the best objective, median, average, and standard deviation values. Using Kolmogorov-Smirnov test, we detect that the obtained values do not follow a normal distribution, then non parametric statistical tests were applied to determine if there is significant difference between algorithms. Wilcoxon rank-sum test was applied for allowing systematic pairwise comparisons and assessing whether one of two samples of independent observations came from populations with the same median. MWPT-SA and MWPT-SA-2P-RPT were compared, being the the null hypothesis rejected in all cases. The test rejected the null hypothesis of equal medians with p-value less than 0.01 in all cases. Then, MWPT-SA-2P-RPT and MWPT-SA-2P-GPT were compared, and also the the null hypothesis rejected at all cases, showing the best performance of MWPT-SA-2P-GPT over MWPT-SA-2P-RPT. Figures 1 and 2 display another perspective of the algorithms behavior.

The comparison between MWPT-SA-2P-GPT and GPT algorithms can be observed in Figure 3.

It is also important to highlight that MWPT-SA-2P-GPT achieved objective values that exceed the values obtained by GPT by between 14% and 71%. The SA algorithms have best behavior with respect to GPT. The objective values of the solutions obtained by the greedy algorithm have low quality with respect to those found by the SA algorithms. In summary, among the proposed algorithms, MWPT-SA-2P-GPT achieves the best performance, obtaining the highest quality solutions.

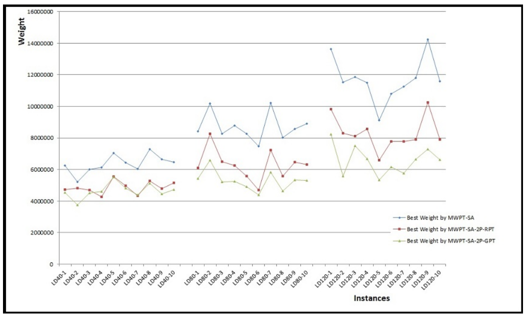

Figure 1: Comparing SA algorithms w.r.t. best values

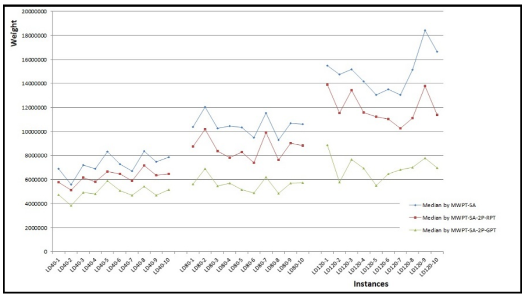

Figure 2: Comparing SA algorithms w.r.t. median values

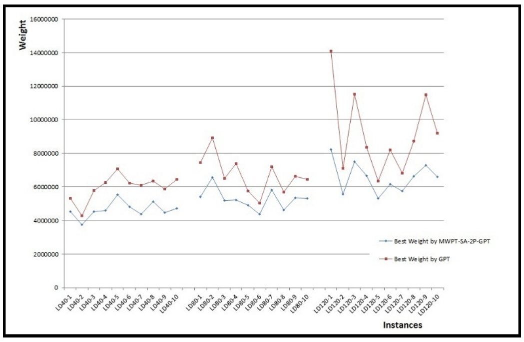

Figure 3: Comparing MWPT-SA-2P-GPT and GPT

In addition, the computational effort of the proposed algorithms applied to the MWPT problem are compared and analyzed. The metaheuristic algorithms consume much more computational resources than a greedy algorithm, but their advantage is that they are capable of achieving solutions of much higher quality. Considering the results obtained and showed in previous subsections for all strategies, Table 1 shows the average runtimes of the mentioned algorithms.

Table 1: Average runtimes of SA algorithms, adding the greedy algorithm for the MWPT problem (in milliseconds).

Instance	MWPT-SA	MWPT-SA-2P-GPT	GPT
40	11679	27210	69
80	25376	33239	83
120	48342	51646	94

4 Conclusions

Our contributions show how SA can be applied to the MWPT problem. We have developed MWPT-SA, MWPT-SA-2P-RPT, and MWPT-SA-2P-GPT algorithms. All claims were corroborated by the experimental study and the respective statical tests. Our conclusions lead us to propose the use of MWPT-SA-2P-GPT for suitably solving MWPT.

References

[1] H. Brönnimann, L. Kettner, M. Pocchiola, and J. Snoeyink. Counting and enumerating pointed pseudotriangulations with the greedy flip algorithm. *SIAM J. Comput.*, 36:721–739, 2006.

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

[3] M. de Berg, M. V. Krevel, M. Overmars, and O. Schwarzkopf. *Computational Geometry. Algorithms and Applications.* Springer-Verlag, 2000.

[4] M. G. Dorzán, E. O. Gagliardi, M. G. Leguizamón, and G. Hernández-Peñalver. Approximations on minimum weight triangulations and minimum weight pseudo-triangulations using ant colony optimization metaheuristic. *Fundamenta Informarticae*, 119(1):1–27, 2012.

[5] E. O. Gagliardi, M. G. Dorzán, M. G. Leguizamón, and G. Hernández-Peñalver. Approximations on minimum weight pseudo-triangulation problem using ant colony optimization. *XXX International Conference of the Chilean Computer Science Society. Jornadas Chilenas de Computación*, 2011.

[6] J. Gudmundsson and C. Levcopoulos. Minimum weight pseudo-triangulations. *Comput. Geom.*, 38(3):139–153, 2007.

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

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

[9] M. Pocchiola and G. Vegter. Pseudo-triangulations: Theory and applications. In *Symposium on Computational Geometry*, pages 291–300, 1996.

[10] G. Rote, F. Santos, and I. Streinu. *Pseudo-triangulations — a survey.* Contemporary Mathematics. American Mathematical Society, December 2008.