# Minimum Average Distance Triangulations


László Kozma

*Advisor*: Raimund Seidel

Universität des Saarlandes, FR 6.2 Informatik, D–66041 Saarbrücken, Germany

kozma@cs.uni-saarland.de


SAARLAND UNIVERSITY

GRADUATE SCHOOL OF COMPUTER SCIENCE

## 1. The general problem

Given $X = \{p_1, \ldots, p_n\}$ **points** in the plane and **weights** $w : X^2 \to \mathbb{R}$, find a geometric, crossing-free graph $T$ embedded on $X$ with edge weights given by $w$, such as to **minimize**:

$$\mathcal{W}(T) = \sum_{1 \le i < j \le n} d_T(p_i, p_j)$$

where $d_T$ is the graph-theoretic distance using $T$.

The solution is always a maximal crossing-free graph, i.e. a **triangulation**. The same question can be asked for vertices of a polygon if we only allow diagonals and boundary edges of the polygon.

We call this problem MAD Triangulation.

## 2. Related problem(s)

Some of the following problems look similar to the MAD Triangulation problem, but we have not found any deep connections:

- Minimum Average Distance Spanning Subgraph in a budgeted version [2] was studied in the context of network design (minimizing average routing time). The problem is NP-complete even with unit weights.
- Minimum Average Distance Spanning Tree [1]. NP-completeness is implied by the previous result.
- In chemistry $\mathcal{W}(T)$ is known as Wiener-index [3] and if it is computed for molecular structures, it correlates with chemical properties of materials [4]. There has been significant research on efficiently computing $\mathcal{W}(T)$ for special graphs [5,6] and on combinatorial properties of $\mathcal{W}(T)$ [7] when edges have unit weight.
- Minimum Weight Triangulation - known to be NP-hard [8].
- Minimum Dilation Triangulation - known to be NP-hard [9].

## 3. The easy problem

All weights are equal to 1 (link distance).

*one-vertex visible polygon*: one of the vertices can see all the other vertices. The set of such polygons forms a subset of *star-shaped* and a superset of *convex* polygons.

*one-point visible set*: one of the points can see all the other points. This is less restrictive than the usual *general position* requirement (no three points collinear).

**Theorem:** For one-vertex visible polygons the solution is the fan. For one-point visible point sets the solution is the extended fan (Figure 1). □

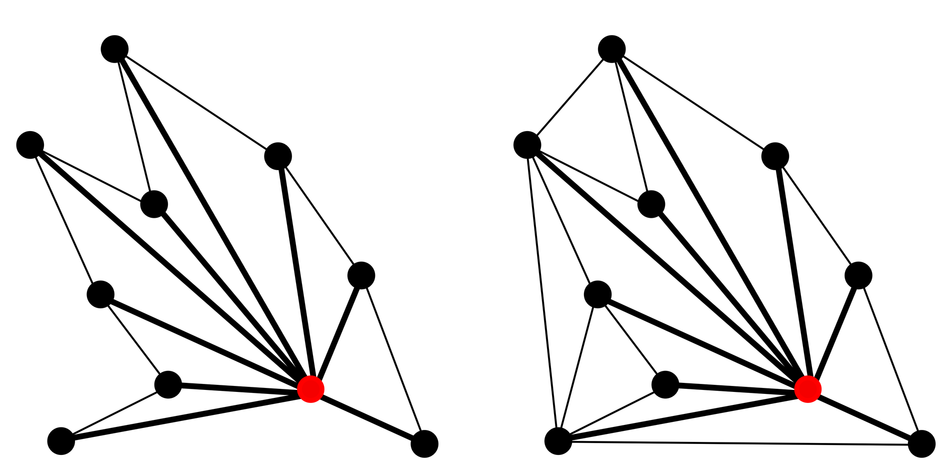

Figure 1: *(a) fan triangulation of a polygon* *(b) extended fan triangulation of a point set*

## 4. The (polynomially) solvable problem

Arbitrary simple polygon (not one-vertex-visible) with all weights equal to 1.

**Idea:** use dynamic programming and split the polygon in two with a triangle that has one side on the boundary (Figure 2).

**Challenge:** How to decompose the cost?

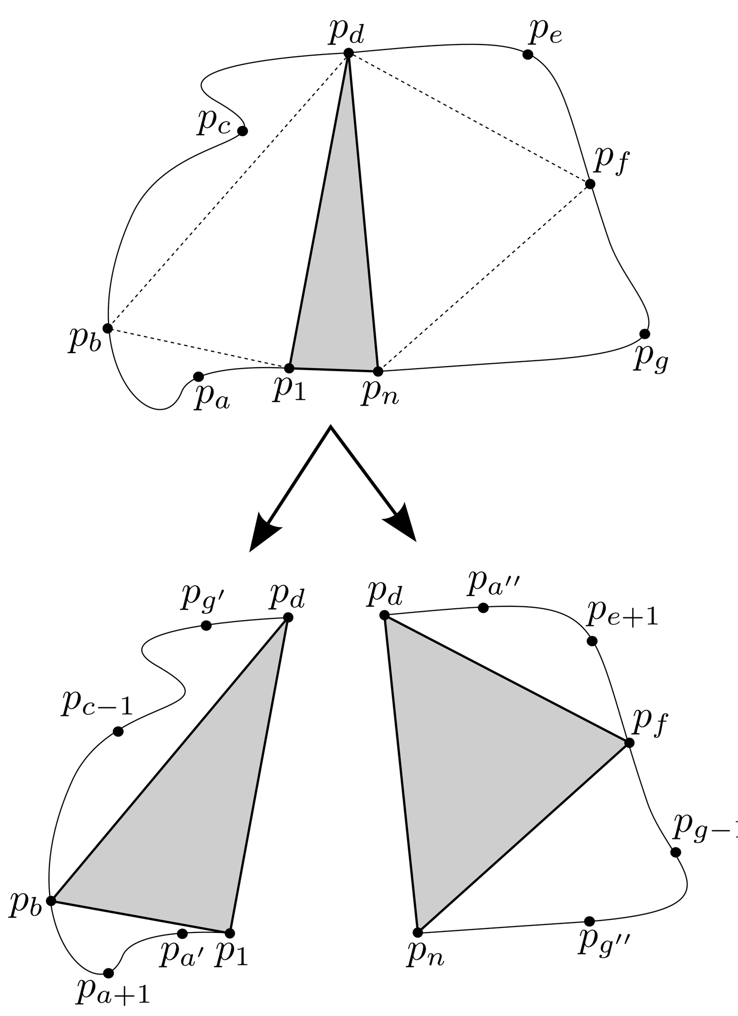


Figure 2: *Simple polygon with special vertices before and after the split*

### Lemma 1 (Special vertices)

Assume $T$ has been found. Visit the vertices in clockwise order from 1 to $n$. Let $p_a$ be the last vertex before $d$ such that $d_T(p_a, p_1) < d_T(p_a, p_d)$ and $p_c$ be the first vertex such that $d_T(p_c, p_d) < d_T(p_c, p_1)$. Let $p_b$ be the other vertex (besides $p_n$) that is connected to both $p_1$ and $p_d$. Then for $k \in [1, d]$:

(i) $d_T(p_k, p_1) < d_T(p_k, p_d)$ iff $k \in [1, a]$;
(ii) $d_T(p_k, p_1) > d_T(p_k, p_d)$ iff $k \in [c, d]$;
(iii) $d_T(p_k, p_1) = d_T(p_k, p_d)$ iff $k \in (a, c)$. In particular: $a < b < c$.

Similarly for $p_d$, $p_e$, $p_f$, $p_g$, $p_n$.

### Lemma 2 (Splitting global distances)

Let $x \in [1, d]$ and $y \in [d, n]$. Let $\phi = d_T(p_x, p_d) + d_T(p_y, p_n)$.
Then $d_T(p_x, p_y)$ can be written in terms of $\phi$, depending on the location of $x$ and $y$, so it can be split into two distances that are locally computable.

$$d_T(p_x, p_y) = \begin{cases} \phi - 1 & \text{if } y \in [d, e]; \\ \phi + 1 & \text{if } y \in [g, n] \text{ and } x \in (a, d]; \\ \phi & \text{otherwise.} \end{cases}$$

### Lemma 3 (Consistency of constraints)

Ignoring the case when $p_1 p_d$ or $p_d p_n$ are on the boundary (in which case the constraints are always observed):

(i) $a$ is the largest index in $[1, d]$ such that $d_T(p_a, p_1) < d_T(p_a, p_d)$ iff $a + 1$ is the smallest index in $[1, b]$ such that $d_T(p_{a+1}, p_b) < d_T(p_{a+1}, p_1)$.
(ii) $c$ is the smallest index in $[1, d]$ such that $d_T(p_c, p_d) < d_T(p_c, p_i)$ iff $c - 1$ is the largest index in $[b, d]$ such that $d_T(p_{c-1}, p_b) < d_T(p_{c-1}, p_d)$.

Similarly on the other side. □

**Solution:** formulate an extended cost function with parameter $\alpha$ (minimizing it with $\alpha = 0$ solves the initial problem). Then write the extended cost recursively in terms of the smaller polygons.

$$\mathcal{W}_{EXT}(T, \alpha)\Big|_1^n = \sum_{1 \le i < j \le n} d_T(p_i, p_j) + \alpha \sum_{1 \le i \le n} d_T(p_i, p_n)$$

$$\mathcal{W}_{EXT}(T, \alpha)\Big|_1^n = \mathcal{W}_{EXT}(T, \alpha_1)\Big|_1^d + \mathcal{W}_{EXT}(T, \alpha_2)\Big|_d^n + \beta$$

Using Lemma 1 and 2 we split the sums until we can identify the two sides and compute $\alpha_1$, $\alpha_2$ and $\beta$ in terms of $\alpha$ and the indices of the special points. Putting it all together:

**procedure** EXT $\left((p_i, \ldots, p_j),\ p_a, p_c, p_e, p_g,\ \alpha\right)$:
  **if** $j = i + 2$:  *(the polygon has only three vertices)*
    **return** $3 + 2\alpha$;
  **else**:
    **return**
$$\min_{\substack{p_d, p'_a, p'_g, p''_a, p''_g: \\ i \le a' \le a+1 \le c-1 \le g' \le d \\ d \le a'' \le e+1 \le g-1 \le g'' \le j \\ p_i \leftrightarrow p_d \leftrightarrow p_j}} \Big\{ \text{EXT}\left((p_i, \ldots, p_d),\ p'_a, p_{a+1}, p_{c-1}, p'_g,\ j - d + \alpha\right) + \text{EXT}\left((p_d, \ldots, p_j),\ p''_a, p_{e+1}, p_{g-1}, p''_g,\ d - i + \alpha\right) + (\alpha + j - g + 1)(d - a - 1) + (e - d + 1)(i - d) \Big\};$$

**Theorem:** The above procedure finds the solution in $O(n^{11})$ time. □

**Note:** the recursion can be stopped when the last vertex in a polygon can see all other vertices. In this case the fan is the optimum, regardless of $\alpha$. This can speed up the process on many instances but it is more difficult to analyze.

## 5. The (NP) hard problem

Point set or simple polygon, weights fulfill the following conditions:

$$\begin{cases} w(p_i, p_j) \ge 0 \\ w(p_i, p_j) = 0 \text{ iff } i = j \\ w(p_i, p_j) = w(p_j, p_i) \\ \text{triangle inequality is **not** necessarily enforced.} \end{cases}$$

The decision problem is:
*For given $\mathcal{W}^\star \in \mathbb{R}$, is there a triangulation $T$ of a given point set $X$ with weights $w$, such that $\mathcal{W}(T) < \mathcal{W}^\star$*

The problem is clearly in NP, as for a given triangulation $T$, we can use an all-pairs shortest path algorithm to compute $\mathcal{W}(T)$ and compare it with $\mathcal{W}^\star$ in polynomial time.

We prove NP-hardness using a reduction from PLANAR 3SAT [10], with some extra restrictions on the admissible formulae. The reduction relies on several gadgets and a lengthy argument.

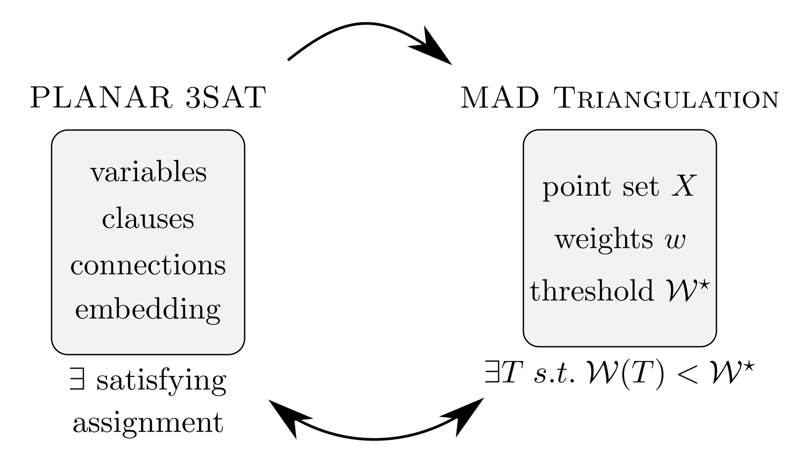


Figure 3: *Reduction from* PLANAR 3SAT

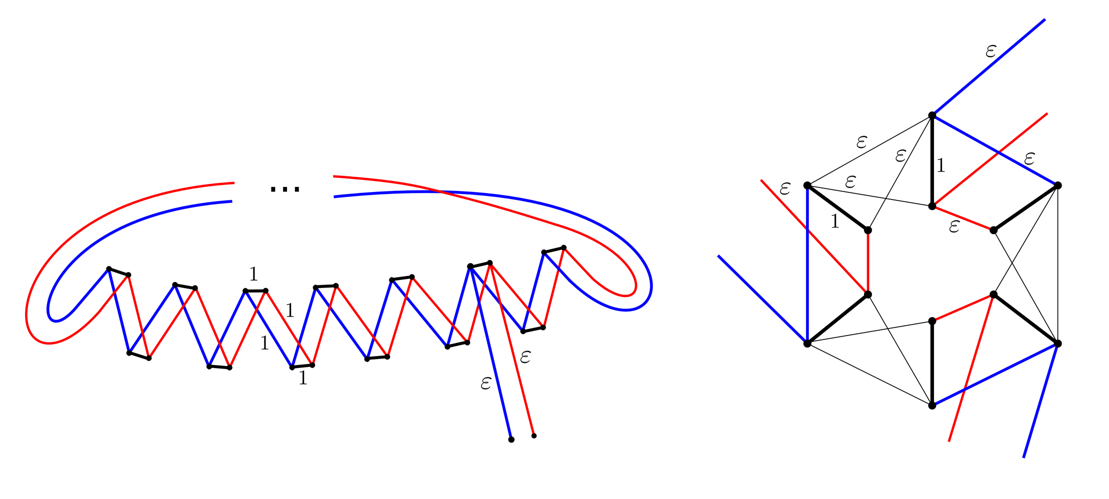


Figure 4: *Gadgets: Variable (left); Clause (right);*

**Theorem:** MAD Triangulation with arbitrary positive weights is NP-Complete. □

## 6. Open problem(s)

(1) Does the problem remain NP-hard if the weights form a metric, in particular the Euclidean metric? What about special cases such as regular polygons or grid points with Euclidean distance?

(2) What is the status of the problem with unit weights for point sets without one-vertex-visibility, e.g. grid points?

(3) Are there good approximations for the hard variants of the problem? For the polynomial case can the running time be improved or more tightly bounded?

(4) Other variants: only integer weights allowed, negative weights allowed, Steiner points, maximization problem, constraints on the allowed graphs besides planarity (total budget on the sum of weights, bounded degree, etc.)

## 7. References


1. T. C. Hu. Optimum communication spanning trees. (1974)
2. D. S. Johnson, J. K. Lenstra et al. The complexity of the network design problem. (1978)
3. H. Wiener. Structural Determination of Paraffin Boiling Points. (1947)
4. D. H. Rouvray. Predicting chemistry from topology. (1986)
5. B. Mohar and T. Pisanski. How to compute the Wiener index of a graph. (1988)
6. C. W. Nilsen. Wiener index and diameter of a planar graph in subquadratic time. (2009)
7. A. Dobrynin, R. Entringer and I. Gutman. Wiener index of trees: Theory and applications. (2001)
8. W. Mulzer and G. Rote. Minimum-weight triangulation is NP-hard. (2008)
9. C. Knauer and W. Mulzer. Minimum dilation triangulations. (2005)
10. D. Lichtenstein. Planar formulae and their uses. (1982)



**Acknowledgement:** We thank Sascha Parduhn for helpful discussions.


❖ ❖ ❖