Dilation of regular polygons

Algorithmic aspects

Damien Galant

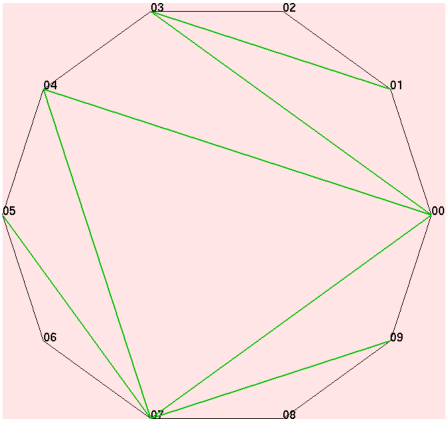
UMONS - Erasmus Université Paris-Sud

October 29, 2019

Introduction and notations

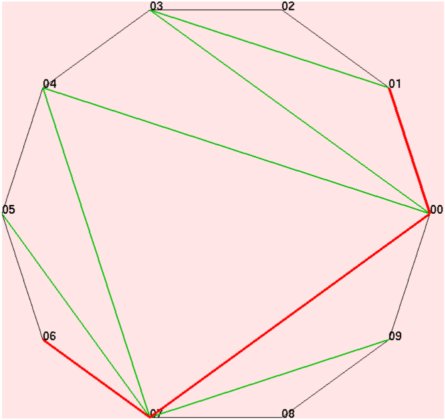
- focus on *regular n-gons*
- S: set of vertices of a regular n-gon
- triangulation on S: maximal set of segments whose endpoints are in S and which only intersect at points of S
- $\mathcal{T}$: set of triangulations of S
- dilation of $T \in \mathcal{T}$: $\mathrm{dil}(T) \geq 1$

Example of a triangulation



A triangulation T of a 10-gon. Corresponding dilation: $\mathrm{dil}(T) = 1.42705098$

Example of a triangulation



The path between a critical pair for this triangulation is shown in red.

$$\mathrm{dil}(T) = \frac{\text{total length of the red path}}{\text{euclidean distance between the endpoints}}$$

What are we looking for?

Computing the dilation of regular *n*-gons, i.e.

$$\min_{T \in \mathcal{T}} \mathrm{dil}(T)$$

For a given $T \in \mathcal{T}$:

- Computing shortest paths in the graph: $O(n^3)$ using Floyd-Warshall's algorithm.
- Iterate over all pairs of points in $O(n^2)$ to get $\mathrm{dil}(T)$.

$\rightarrow O(n^3)$ overall

Combinatorial explosion

- Straightforward algorithm: iterate over all possible triangulations T (see e.g. Mulzer (2004)).
- Impossible for $n \geq 25$: the number of triangulations of a n-gon is equal to the Catalan number C_{n-2}, where

$$C_k = \frac{1}{k+1} \cdot \binom{2k}{k}$$

($C_{23} = 343.059.613.650$)
$\rightarrow$ **combinatorial explosion**

Proposed solution

- "Branch-and-bound-like" approach.
- Lower bound method: inspired by Dumitrescu and Ghosh (2016).

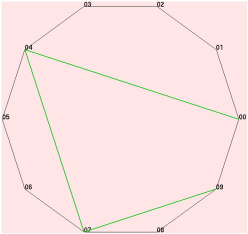
Lower bound: what are we looking for?

- We want a *proven* lower bound for the dilation of regular n-gons.
- If the found lower bound can be realized as $\mathrm{dil}(T)$ for some $T \in \mathcal{T}$, we are done.

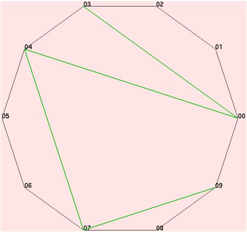
Partial triangulations

- Partial triangulation: set of segments whose endpoints are in S and which only intersect at points of S (no maximality condition).
- We consider that the edges of the polygon are always present.
- $\mathcal{P}$: set of (possibly) partial triangulations.
- Natural notion of inclusion $P_1 \subset P_2$ for $P_1, P_2 \in \mathcal{P}$.

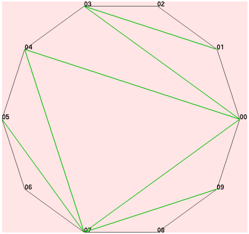
Examples of (partial) triangulations



P_1



P_2



P_3

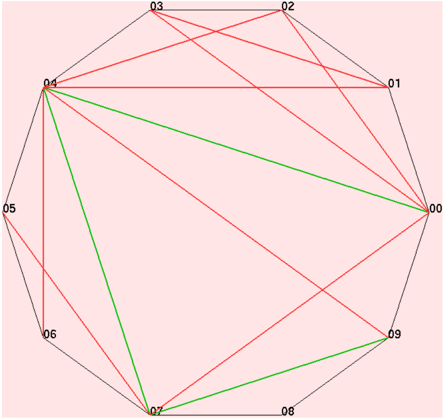
$$P_1 \subset P_2 \subset P_3$$

$$P_3 \in \mathcal{T}$$

Graphs with cliques

- Given P, we are interested in *all* triangulations containing P.
- The graph GC_P is obtained by taking all segments between points of S which do not intersect segments of P.
- "Duality": for $T \in \mathcal{T}$, $P \subseteq T \Leftrightarrow T \subseteq GC_P$

A graph with cliques GC_P



10-gon, three segments in P (shown in green), GC_P: green and red segments
$\mathrm{nlb}(P) = 1.42705098$

Lower bound from a partial configuration

- Given P, “naive” lower bound on the dilation of *all* triangulations containing P given by

$$\mathrm{nlb}(P) := \max_{\substack{p,q \in S \\ p \neq q}} \frac{d_{GC_P}(p,q)}{d_{\text{Euclidean}}(p,q)}$$

- Monotonicity:

$$P \subseteq P' \Rightarrow \mathrm{nlb}(P) \leq \mathrm{nlb}(P')$$

- If $T \in \mathcal{T}$ is a triangulation,

$$\mathrm{nlb}(T) = \mathrm{dil}(T)$$

Summary of the “naive” lower bound technique

$$
\begin{array}{lcl}
P & \rightarrow & \text{partial triangulation} \\
\Downarrow & & \\
GC_P & \rightarrow & \text{add all segments which don't intersect } P \\
\Downarrow & & \\
d_{GC_P} & \rightarrow & \text{distance using only segments in } GC_P \\
\Downarrow & & \\
\mathrm{nlb}(P) & \rightarrow & \text{“naive” lower bound from } P
\end{array}
$$

The lower bound technique

- We want a better bound $\mathrm{lb}(P)$ with

$$\mathrm{nlb}(P) \leq \mathrm{lb}(P) \leq \min_{\substack{T \in \mathcal{T} \\ P \subseteq T}} \mathrm{dil}(T)$$

- We use GC_P (as for nlb).

Pairs of pairs of points

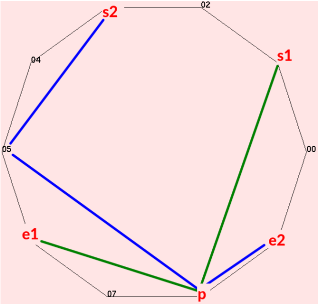
- Idea of nlb: use the inequality

$$d_{GC_P}(p,q) \leq d_{\text{Graph of } T}(p,q)$$

  for a fixed pair of points $p, q \in S, p \neq q$.
- Problem: *pairs of points are considered independently.*
- Solution (inspired by Dumitrescu and Ghosh (2016)): *consider two pairs of points at once.*

Pairs of pairs of points

Simple observation: if $s_1, s_2, e_1, e_2 \in S$ are distinct points in clockwise order, then the paths from s_1 to e_1 and from s_2 to e_2 must intersect at some point $p \in S$.



Pairs of pairs of points

- We have no idea of which p is optimal $\rightarrow$ take the one which gives the lowest bound.
- The bound $\mathrm{lb}(s_1, s_2, e_1, e_2)$ associated to $s_1, s_2, e_1, e_2 \in S$ is

$$\min_{p \in S} \max \left\{ \frac{d_{GC_P}(s_1, p) + d_{GC_P}(p, e_1)}{d_{\text{Euclidean}}(s_1, e_1)}, \frac{d_{GC_P}(s_2, p) + d_{GC_P}(p, e_2)}{d_{\text{Euclidean}}(s_2, e_2)} \right\}$$

- We obtain our better bound

$$\mathrm{lb}(P) = \max_{\substack{s_1, s_2, e_1, e_2 \in S \\ \text{distinct and} \\ \text{in clockwise order}}} \mathrm{lb}(s_1, s_2, e_1, e_2)$$

What we have and what we want

- Lower bound technique: lower bound $\mathrm{lb}(P)$ on the dilation of triangulations which contain P.
- Our goal: find a global lower bound glb with

$$\mathrm{glb} \leq \min_{T \in \mathcal{T}} \mathrm{dil}(T)$$

and a sharp inequality ("$=$" $\rightarrow$ dil computed).

Algorithm to find a global lower bound

- Algorithm for glb: take

$$\mathrm{glb} = \min_{P \in \mathcal{C}} \mathrm{lb}(P)$$

  where $\mathcal{C} \subseteq \mathcal{P}$ is a set of partial configurations.
- Exhaustive method: case $\mathcal{C} = \mathcal{T}$!

Global lower bound: which configurations should we consider?

- How does the algorithm choose $\mathcal{C}$?
- Key point: good tradeoff between $\mathcal{C}$ small (fast algo, possibly poor bound) and $\mathcal{C}$ large (slower, better bound).

The search tree

- Abstract "search tree" of partial configurations $P \in \mathcal{P}$.
- For each P, we have a bound $\mathrm{lb}(P)$.
- Monotonicity is important: if $P_0 \subseteq P_1 \subseteq \cdots \subseteq P_n = T \in \mathcal{T}$, then

$$\mathrm{lb}(P_0) \leq \mathrm{lb}(P_1) \leq \cdots \leq \mathrm{lb}(P_k) = \mathrm{dil}(T)$$

Pruning the search tree

Pruning is very efficient for optimisation problems on search trees
$\rightarrow$ need a "target value"

Lower bound, with a "target value" c

Given a constant

$$c \geq \min_{T \in \mathcal{T}} \mathrm{dil}(T)$$

return a *proven* lower bound

$$\mathrm{glb} \leq \min_{T \in \mathcal{T}} \mathrm{dil}(T)$$

In practice, $c = \mathrm{dil}(T_{candidate}) \in \mathcal{T}$ for a very good triangulation T.

What is c useful for?

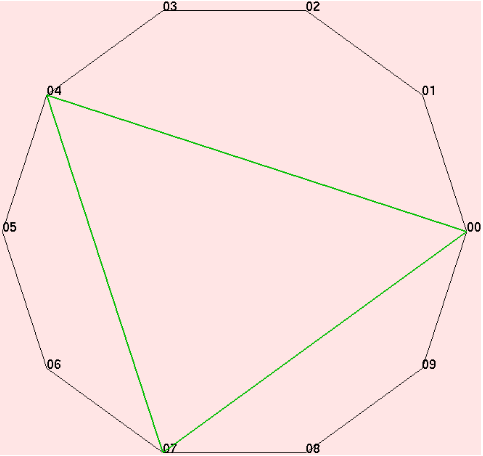
- c is only used for pruning purposes
  $\rightarrow$ "cut" branches of the search tree
- c, given as input to the lower bound algorithm, *does not change the result returned by the algorithm* (!)
- The speed of the proposed method depends *crucially* on the "quality" of c.
- Hope: prove that c is in fact equal to the dilation, i.e.

$$\mathrm{glb} = c = \mathrm{dil}(T_{candidate})$$

Important edges first

- The order in which partial configurations are considered matters.
- Important to first put some edges that will cause $\mathrm{lb}(P)$ to be big, to cut early.
- Our program puts the *edges of the triangle which contains the center* first.
- It then puts three smaller triangles on the 3 zones delimitated by the central triangle.

Central triangle



A possible central triangle in a 10-gon.

Putting it all together

Lower bound algorithm

1. Take a positive integer n and a "target value" c as input.
2. Go through the search tree of partial triangulations, considering important edges first (adding triangles gradually).
3. Prune while going through the search tree.
4. Stop at a specified depth.
5. Return the global lower bound glb.

Upper bound: what are we looking for?

As we saw before, we need a good target constant $c = \mathrm{dil}(T_{good})$ if we want our lower bound algorithm to run fast enough, and we can only conclude if

$$c = \min_{T \in \mathcal{T}} \mathrm{dil}(T)$$

Classical techniques

- Most articles only focus on the upper bound part: find T_{good},

$$\min_{T \in \mathcal{T}} \mathrm{dil}(T) \lessapprox \mathrm{dil}(T_{good})$$

- Two typical steps:
  1. Describe a class of "seemingly good" triangulations (classes with 4 and 6 parameters in Sattari and Izadi (2019)).
  2. Find the optimal triangulation among the members of the class.

Discussion of such techniques

Two main advantages:

- The number of considered configurations is polynomial in n.
- Finding the best configuration
  $\rightarrow$ doable either with a computer or by hand.

Intrinsic issues:

- No formal justification regarding why these classes are considered, only heuristic motivations.
- *(!) No control on the sharpness of the inequality*

$$\min_{T \in \mathcal{T}} \mathrm{dil}(T) \leq \mathrm{dil}(T_{good})$$

Discussion of such techniques

- Second issue: due to the nature of the methods, i.e. living in $\mathcal{S} \subseteq \mathcal{T}$ and forgetting about the rest of $\mathcal{T}$.
- Lower bound algorithm $\rightarrow$ response to the second issue.
- To avoid these issues, we will use *metaheuristics* instead to find good configurations.

Metaheuristics

- Goal: explore the search space $\mathcal{T}$ and find good configurations.
- Metaheuristics: generic methods to solve optimization problems.
- Here: *hill climbing*.

Hill climbing

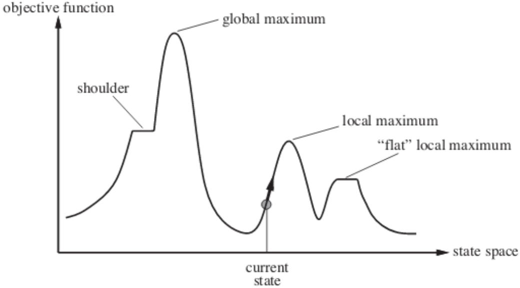
Given "neighbourhood operations" on the search space:

Hill climbing

1. Start from some initial state s_0 in the configuration space.
2. Consider all neighbours of s_0.
3. Go to the neighbour which corresponds to the highest value.
4. When all neighbours produce a lower value, stop the algorithm and return the current state and the current value.

From local maxima to candidates of global maxima

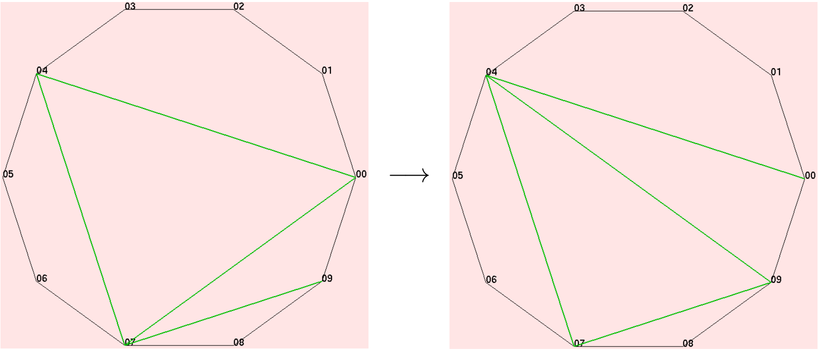
Hill climbing → *local* maxima.



Source: https://www.geeksforgeeks.org/introduction-hill-climbing-artificial-intelligence/

Solution → "randomized multistart" strategy.

An example of neighbourhood operation



Example of 42-gons

Let's do it live!

Known values for the dilation before our work

n	$\mathrm{dil}(S_n)$	n	$\mathrm{dil}(S_n)$	n	$\mathrm{dil}(S_n)$
4	1.4142	12	1.3836	20	1.4142
5	1.2360	13	1.3912	21	1.4161
6	1.3660	14	1.4053	22	1.4047
7	1.3351	15	1.4089	**23**	**1.4308**
8	1.4142	16	1.4092	24	1.4013
9	1.3472	17	1.4084	25	< 1.4296
10	1.3968	18	1.3816	26	< 1.4202
11	1.3770	19	1.4098		

The values of $\mathrm{dil}(S_n)$ for $n = 4, \ldots, 26$, from Dumitrescu and Ghosh (2016).

New exact values computed by our algorithm

n	$\mathrm{dil}(S_n)$	time	n	$\mathrm{dil}(S_n)$	time	n	$\mathrm{dil}(S_n)$	time
20	1.4142	< 5s	28	1.4147	20s	36	?	–
21	1.4161	< 5s	29	1.4198	< 10s	37	?	–
22	1.4047	< 5s	30	1.4236	2min	38	1.4130	1min
23	1.4308	< 5s	31	1.4119	1min	39	?	–
24	1.4013	< 5s	32	1.4160	20s	40	?	–
25	1.4049	15s	33	1.4184	2min	41	?	–
26	1.4169	15s	34	1.4167	1min	42	1.4222	15s
27	1.4185	15s	35	1.4212	3min	43	1.4307	3min

The values of $\mathrm{dil}(S_n)$ computed by our programs, with the associated total runtime (upper bound + lower bound).

Maximal dilation of a convex polygon

- Our method gives (after approximately 30min)

$$\mathrm{dil}(\text{53-gons}) \geq 1.4336$$

- This improves the bound of $\mathrm{dil}(\text{23-gons}) \approx 1.4308$ obtained in Dumitrescu and Ghosh (2016) for the "worst-case dilation of plane spanners":

$$\sup_{\substack{S \subseteq \mathbb{R}^2 \\ S \text{ finite}}} \mathrm{dil}(S)$$

Further goals

- Study the asymptotic case, i.e. the *dilation of the circle*.
- Find "small" classes containing optimal configurations.
- Finer information about small configurations: all good configurations, their symmetries, ...
- Perhaps a "real branch-and-bound" instead of our "two-steps" method.

Bibliography

Adrian Dumitrescu and Anirban Ghosh. "Lower Bounds on the Dilation of Plane Spanners". In: *Algorithms and Discrete Applied Mathematics*. Ed. by Sathish Govindarajan and Anil Maheshwari. Cham: Springer International Publishing, 2016, pp. 139–151. ISBN: 978-3-319-29221-2.

Wolfgang Mulzer. "Minimum dilation triangulations for the regular n-gon". MA thesis. 2004. URL: https://page.mi.fu-berlin.de/mulzer/pubs/diplom.pdf.

Sattar Sattari and Mohammad Izadi. "An improved upper bound on dilation of regular polygons". In: *Computational Geometry* 80 (2019), pp. 53–68. ISSN: 0925-7721. DOI: https://doi.org/10.1016/j.comgeo.2019.01.009.