Dilation of regular polygons

Algorithmic aspects (detailed version)

Damien Galant

UMONS - Erasmus Université Paris-Sud

October 29, 2019

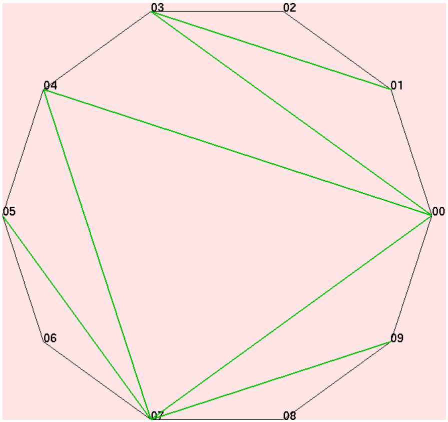
Introduction and notations

In this talk, we will focus on the dilation of regular n-gons.
However, most of what will follow (including the algorithms) can be generalized easily, at least to strictly convex polygons.
We will denote by S the set of points of a regular n-gon in $\mathbb{R}^2$, say

$$S = \left\{ \left(\cos\left(\frac{2k\pi}{n}\right), \sin\left(\frac{2k\pi}{n}\right) \right) \middle| 0 \leq k < n \right\}$$

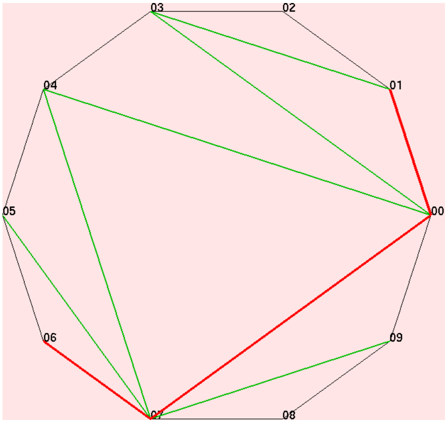
A triangulation on S is a maximal set of segments whose endpoints are in S and which only intersect at points of S.
We will denote by $\mathcal{T}$ the (finite) set of triangulations of S.
Given $T \in \mathcal{T}$, we will denote by $\mathrm{dil}(T) \geq 1$ the dilation of T.

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?

Given n, we are interested in computing the dilation of regular n-gons, i.e.

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

Given T, computing $\operatorname{dil}(T)$ can be done in $O(n^3)$ using Floyd-Warshall's algorithm to compute "all pairs shortest paths". We can then iterate over all pairs of points in $O(n^2)$ and compute the dilation of T.
The overall complexity is therefore $O(n^3)$.

Combinatorial explosion

A straightforward way to compute the dilation of regular n-gons is to iterate over all possible triangulations T (up to some symmetries to speed-up the computations).
This was done for instance in **mulzer2004**.
We cannot hope to compute dilations of n-gons even for $n \geq 25$ using this method since the number of triangulations of a n-gon is given by the $(n-2)$th Catalan number C_{n-2}, where

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

$(C_{23} = 343.059.613.650)$
This implies a combinatorial explosion of the number of triangulations as n grows, therefore ruling out purely "bruteforce" approaches to compute the dilation of regular n-gons.

Proposed solution

In this talk, we will present a “branch-and-bound-like” approach to compute the dilation of regular *n*-gons.
The lower bound method was inspired by a technique used in **dumitrescu2016** to compute the dilation of regular 23-gons.

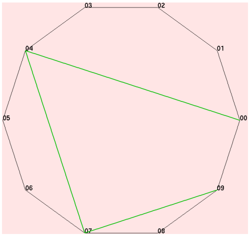
Lower bound: what are we looking for?

We are going to present an algorithm which, given *n*, returns a *proven* lower bound for the dilation of regular *n*-gons.
If the found lower bound can be realized by a suitable configuration, then this value will be equal to the dilation of regular *n*-gons and we are done.

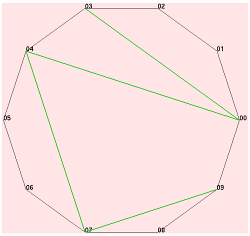
Partial triangulations

By a partial triangulation, we mean a set of segments whose endpoints are in S and which only intersect at points of S (we remove the maximality condition from the definition of triangulations). However, we assume that the edges of the polygons are always present in our configurations.
Let's denote the set of (possibly) partial triangulations by $\mathcal{P}$.
Inclusion of partial triangulations is defined in the natural way.

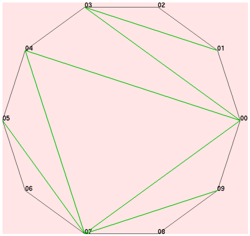
Examples of (partial) triangulations



P_1



P_2



P_3

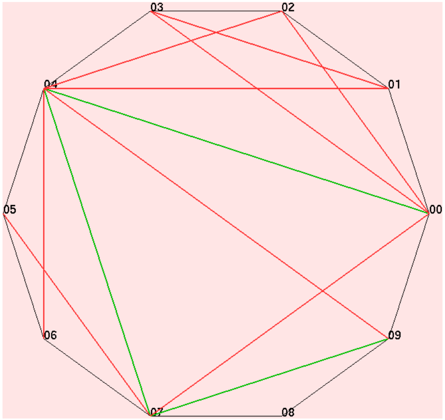
$$P_1 \subset P_2 \subset P_3$$

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

Graphs with cliques

- Given a partial triangulation $P \in \mathcal{P}$, 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.
- There is a kind of "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 a partial triangulation P, a "naive" lower bound on the dilation of *all* triangulations containing P is given by

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

There is a monotonicity property:

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

Futhermore, $\mathrm{nlb}(T) = \mathrm{dil}(T)$ if $T \in \mathcal{T}$ (i.e. the bound is exact for maximal elements of $\mathcal{P}$)

Summary of the "naive" lower bound technique

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

The lower bound technique

We want to find a better bound, i.e. a value $\mathrm{lb}(P)$ such that

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

Since all edges that do not intersect an edge from P will occur in at least one configuration $T \in \mathcal{T}$ such that $P \subseteq T$, we will use the "graph with cliques" GC_P just as for nlb.

The problem can also be seen like this: from the "graph with cliques" GC_P associated to P, find a bound

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

(this is a kind of "dual problem").

Indeed, for $T \in \mathcal{T}$ we have $P \subseteq T \Leftrightarrow T \subseteq GC_P$ by "duality".

Pairs of pairs of points

The dilation of a triangulation $T \in \mathcal{T}$ is given by

$$\max_{\substack{p,q \in S \\ p \neq q}} \frac{d_{\text{Graph of } T}(p,q)}{d_{\text{Euclidean}}(p,q)}$$

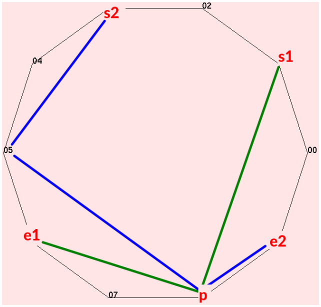
The naive lower bound considers a single pair of points $p, q \in S, p \neq q$ and uses the inequality (where $P \in \mathcal{P}$, $P \subseteq T$)

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

This is not optimal since pairs of points are considered independently. Instead, the new method considers *two pairs of points at once* (this idea was inspired by **dumitrescu2016**).

Pairs of pairs of points

We begin with a 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

Since we have no idea of which p is optimal, we take the one which gives the lowest bound for the pair of pairs.
In the end, 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 finally obtain

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

Global lower bound

The lower bound technique provides a lower bound $\mathrm{lb}(P)$ on the dilation of triangulations which contain P.
Our goal is to find a good (unique) global lower bound glb with

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

with an inequality as sharp as possible (with equality if possible, since this would mean that glb is the exact value of the dilation).
Our algorithm will take

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

where $\mathcal{C} \subseteq \mathcal{P}$ is a set of partial configurations such that

$$\forall T \in \mathcal{T}, \exists P \in \mathcal{C}, P \subseteq T$$

Note that the exhaustive method corresponds exactly to the case $\mathcal{C} = \mathcal{T}$!

Global lower bound: proof of correctness

glb satisfies

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

Indeed if $T \in \mathcal{T}$ then, by hypothesis on $\mathcal{C}$, there exists $P_T \in \mathcal{C}$ such that $P_T \subseteq T$. We then have

$$\mathrm{glb} = \min_{P \in \mathcal{C}} \mathrm{lb}(P) \leq \mathrm{lb}(P_T) \leq \mathrm{lb}(T) = \mathrm{dil}(T)$$

Global lower bound: which configurations should we consider?

What remains to be done is to explain how the algorithm chooses $\mathcal{C}$, the set of partial configurations to be considered.
The key point is to find a good tradeoff between having $\mathcal{C}$ small, with a fast algorithm but a possibly poor bound, and $\mathcal{C}$ large, with a slower method but a better bound (maybe optimal, if $\mathcal{C}$ is large enough)

The search tree

In the end, we have an abstract "search tree" of partial configurations (the lattice of sets [1] of $\mathcal{P}$, ordered by inclusion) and for each $P \in \mathcal{P}$, we have a bound $\mathrm{lb}(P)$.
Recall that, if $P_0 \subseteq P_1 \subseteq \cdots \subseteq P_n = T \in \mathcal{T}$, then

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

[1]Which will be abusively called the "search tree", even though one might argue that it is not really a tree.

Pruning the search tree

A very common and efficient technique to deal with optimisation problems on search trees is *pruning*, i.e. stop exploring branches of the tree which do not lead to optimal solutions.
We therefore provide a “target value” to our lower bound algorithm:

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

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

Typically, we will use $c = \mathrm{dil}(T_{candidate}) \in \mathcal{T}$, the dilation of a possibly optimal triangulation (or at least a very good triangulation).

What is c useful for?

c is only used for pruning purposes: if a partial triangulation $P \in \mathcal{P}$ is considered and

$$\mathrm{lb(P)} \geq c$$

then we know that

$$\forall T \in \mathcal{T} : \left((P \subseteq T) \Rightarrow c \leq \mathrm{lb}(P) \leq \mathrm{lb}(T) = \mathrm{dil}(T) \right)$$

and we can "cut" that branch of the search tree.

A peculiar feature of the lower bound method is that 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. It is therefore important to find good configurations beforehand.

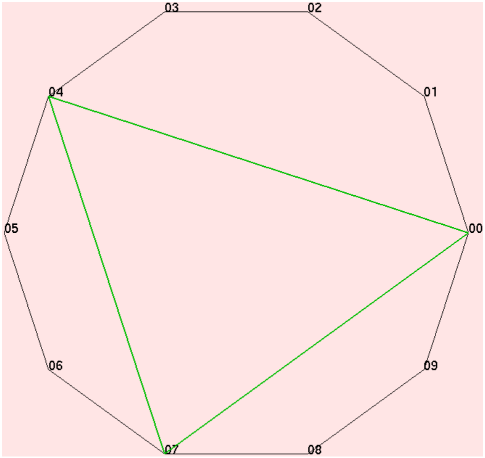
The hope is that the lower bound algorithm will prove that c is in fact equal to the dilation, i.e.

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

Important edges first

- The last (important!) fact we did not mention is the order in which partial configurations are considered.
- Indeed, it is important to first put some edges that will likely cause $\mathrm{lb}(P)$ to be big and hopefully to cut this branch of the search tree.
- In practice, our program puts the *edges of the triangle which contains the center* first.
- It then puts (or tries to put) 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

To summarize, our algorithm works as follows:

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 (or try to do some processing at the leaves of the search tree, which correspond to configurations which were not pruned).
5. Return the global lower bound glb.

If $\mathrm{glb} = c$, we managed to compute the dilation!

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 in the field only focus on the upper bound part, i.e. finding configurations $T_{good} \in \mathcal{T}$ so that

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

with a "rather sharp" inequality, possibly an equality (see for instance **sattari2019**, which was a starting point of our work with C. Pilatte)
Typically, such articles proceed as follows:

1. Consider a class of "seemingly good" triangulations with a few parameters (classes with 4 and 6 parameters were considered in **sattari2019**).
2. Find (somehow) the optimal triangulation among the members of the class. This best triangulation is the candidate T_{good}.

Discussion of such techniques

These techniques have two main advantages:

- The number of considered configurations is polynomial in n, allowing to find bounds even for large values of n.
- Finding the best configuration among the class of considered configurations is doable either with a computer (since the state space has a polynomial size) or even "by hand" due to the specific structure of the considered triangulations.

However, we argue that these approaches also have intrinsic issues:

- Often, there is no formal justification regarding why these classes are considered, but rather heuristic motivations: the class contains the optimal examples for small n, has enough parameters, ...
- *(!) There is 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

- The second issue is really due to the nature of the methods, which consist in living in a (small and better understood) subset

$$\mathcal{S} \subseteq \mathcal{T}$$

  and forgetting about the rest of $\mathcal{T}$.
- The whole point of the lower bound algorithm is to respond to the second issue.
- As we want to investigate what happens for $n \geq 25$ and do not have convincing candidates of polynomial-size subclasses of $\mathcal{T}$ which should contain optimal triangulations, we will use *metaheuristics* instead to find good configurations, among which we hope to find optimal ones.

Metaheuristics

- We want to explore the search space $\mathcal{T}$ and find good configurations.
- Finding maxima [2] of real functions defined on large discrete search spaces is an intensively studied subject, and generic methods to solve these kinds of optimization problems are often called metaheuristics.
- We used one of the most classical approaches: *hill climbing*.

[2] Here we want to find a global minimum but we will stick with the common terminology used for maximization problems to explain briefly what are metaheuristics.

Hill climbing

Given “neighbourhood operations” on the search space, the following algorithm can be used:

Hill climbing

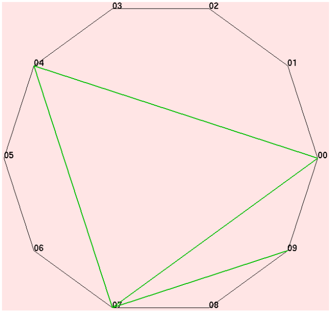
1. Start from some initial state s_0 in the configuration space.
2. Consider all neighbours of s_0.
3. Go to the neighbour which corresponds to the highest value.
4. When all neighbours produce a lower value, stop the algorithm and return the current state and the current value.

From local maxima to candidates of global maxima

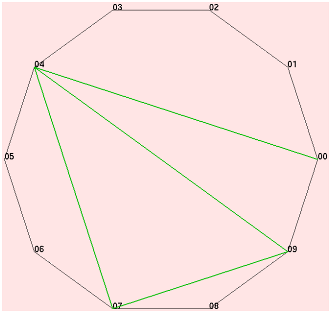
- It is clear that the mentioned algorithm only gives rise to local maxima.
- Generally, the problem of finding the global maximum of a function on a search space is hard without further hypotheses.
- In practice, we used a “randomized multistart” strategy: we run the hill climbing algorithm from a lot of triangulations chosen uniformly at random from $\mathcal{T}$.
- By doing so, we explore different regions of the search space and are less likely to miss the global maxima.

An example of neighbourhood operation

Take a quadrilateral from T and replace its diagonal by the other diagonal.



$\longrightarrow$



To avoid being stuck in local minima, we can use this neighbourhood operation with depth 2.

Example of 42-gons

- In less then 10s, the upper bound program gives a value of 1.4222217969.
- The lower bound program then proves that this bound is correct in less than 3s.

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 **dumitrescu2016**

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).

Maxmial dilation of a convex polygon

- Our lower program shows (after approximately 30min) that the dilation of 53-gons is at least 1.4336430827. We do not know the exact value of the dilation of 53-gons however.
- We thereby improve the bound of $\mathrm{dil}(23\text{-gons}) \approx 1.4308$ obtained in **dumitrescu2016** for the “worst-case dilation of plane spanners”, i.e. the maximal dilation of a set of points:

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

Further goals

- One of the main goals is to use the hindsight given by small cases to study the asymptotic case, i.e. the dilation of the circle.
- An interesting observation would be to find “small” classes containing optimal configurations, therefore going back (but with more support) to classical upper bound methods.
- Our program allows to have finer information about small configurations: find all good configurations, their symmetries, ...
- We might move towards a “real branch-and-bound” instead of our “two-steps” method.

Bibliography