An Efficient approach to finding convex hull of a finite point set in $\mathbb{R}^2$

Candidate Name: Jiajie (Jason) Ma

Candidate Number:

School: High School Affiliated to Shanghai Jiao Tong University

Table of Contents

Introduction

In mathematics, convex is a classic concept that reveals many properties of a geometry in all dimensional spaces. Specifically, in an Euclidean plane, a region is intuitively considered to be convex if all of its interior angles are smaller than $180°$. More generally speaking, it also means a “guard” at any point in the region can “see” every other point in it. In this paper, we will focus on a special kind of convex geometry in $\mathbb{R}^2$ named convex hull that has been deeply researched and applied in many fields including image processing, geographic information system (GIS) and so on. To make an analogy, a group of meerkats are scattering statically on an island and convex hull is the minimal convex region to fence them for specie protection. Furthermore, it’s a fundamental computational geometry problem to find convex hull for a finite point set as input. Hence, in first section, we will formally define convex hull mathematically and also investigate some of its properties. A huge emphasis will then be put on finding an efficient algorithm to computing convex hull of a finite point set in $\mathbb{R}^2$ and prove its accuracy. Finally, computational complexity will be adopted to evaluate the quality of the algorithm and give us a lower bound.

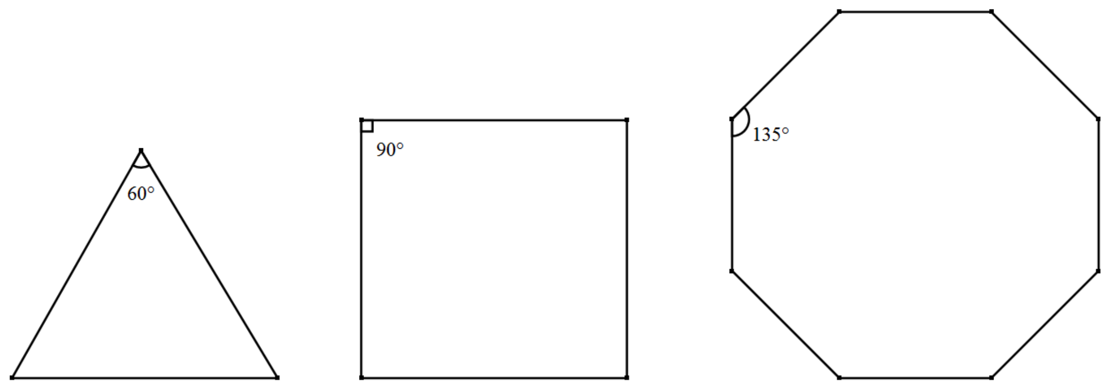

Fig 1. Convex Polygon in $\mathbb{R}^2$

Convex Region

Given a polygon in $\mathbb{R}^2$, such as equilateral triangle, square or even regular octagon, it's intuitive to determine their convexity by checking if all a polygon's interior angles are smaller than 180° — if so, we'll say the shape is convex; otherwise, it's concave. However, when talking about convexity, it's in a narrow sense to only refer it to polygons and take angles as determinant. For instance, the circle surface of a drum is also considered to be convex by intuition, but it's not strictly a polygon. Hence, in this paper, without the loss of generality, we adopt the term "region" to indicate space in $\mathbb{R}^2$ as follow so that we'll develop the idea of convex and convex hull from it:

Def 1. *A **region R** is a space in* $\mathbb{R}^2$.

Furthermore, for the sake of simplicity, by indicating $\mathbb{R}^2$, we assume that all points in it can be expressed by coordinates (x, y).

Also, for generality, here we give and adopt three instinctive criteria in this paper with OR relations to determine if a region R is convex:

(1) **Criteria 1.** A region R is convex if a “guard” at any point in the region can “see” every other points in the region.

(2) **Criteria 2.** A region R is convex if every tangent line of the region doesn’t intersect the region itself.

(3) **Criteria 3.** A region R is convex if for arbitrary two points $x, y \in R$, the line segment $\overline{xy}$ is inside R.

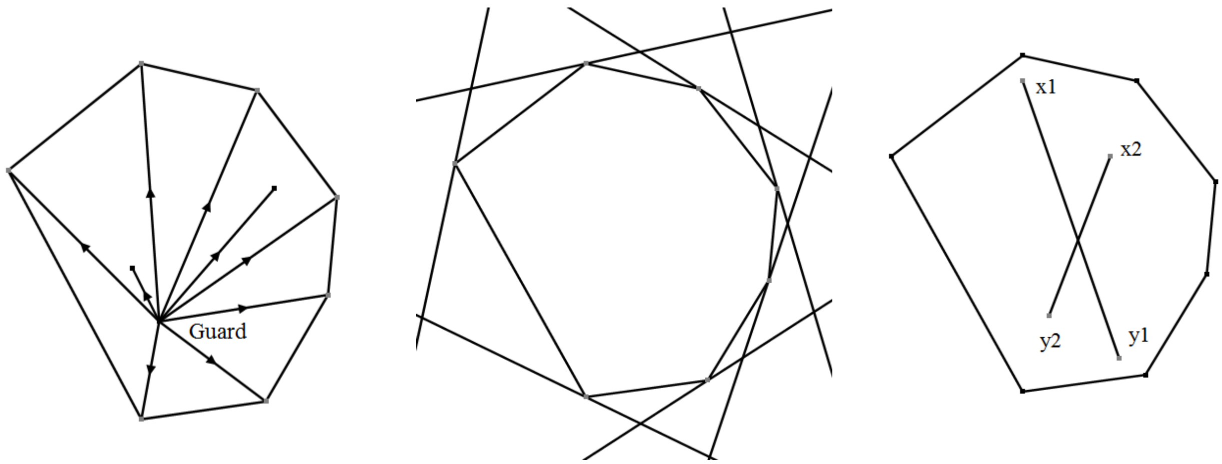

Fig 2. 3 Criteria to determine convexity of a region

Otherwise, the region is concave.

Convex Hull

In a finite region R, we want to hitch n stakes planted on the ground with an elastic ring so that they’re all included and the ring can’t be tighter--it can be easily found that the shape of the ring is now defined by “the most outer” stakes with linear interpolation and looks like a polygon, specially called “convex hull”.

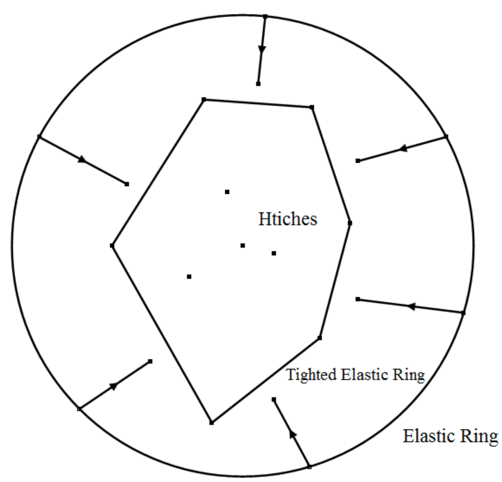

Fig 3. Hitches bounded by an elastic ring

Instinctively, we can see that convex hull (the ring) is a minimal convex region ⊑ R enclosing all n stakes. However, to investigate rigorously, in this section, we'll first define convex hull mathematically and discuss its properties with proposed proofs and implications.

Definition

In this paper, we adopt the general definition of convex hull in $\mathbb{R}^2$ in computational geometry as follows (by MathWorld):

Def 2. *For a set of points,* $S = \{p_1, p_2, p_3 \ldots, p_n\}$, *the convex hull of S,* $\boldsymbol{conv}(\boldsymbol{S})$, *is the intersection of all convex regions containing S.*

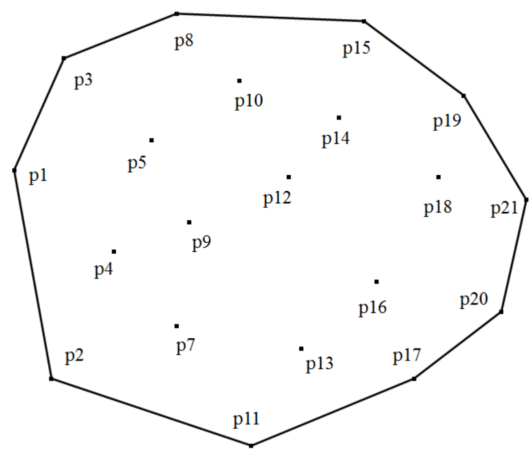

Fig 4. Convex Hull of a finite point set in $\mathbb{R}^2$

Property

Different from the denotation of "convex hull", its formal definition only reveals the "minimum" aspect of the "enclosing" region but doesn't direct to its convexity. Furthermore, the intuitive shape of convex hull — linear interpolation — is motivating to be linked to vector region in linear algebra (like the geometric implication of Simplex algorithm). Hence, in this subsection, we will particularly focus on discussions of convex hull's convexity and its algebraic significance.

Convex

To determine if convex hull is convex, it's straightforward to check if it fits any one of our "Convex region criteria" discussed before. Hence, the proof is presented as follows:

Claim: *Convex Hull is* ***convex***

Proof: By definition of convex hull, consider any two convex regions A, B that contains a finite point set in $\mathbb{R}^2$. We want to show the intersection of A and B, $\mathrm{C} = \mathrm{A} \cap \mathrm{B}$, is convex as well.

Let $\mathrm{x}, \mathrm{y} \in \mathrm{C}$. From $\mathrm{C} = \mathrm{A} \cap \mathrm{B}$, we know $\mathrm{x}, \mathrm{y} \in \mathrm{A}$ and $\mathrm{x}, \mathrm{y} \in \mathrm{B}$.

Furthermore, by **criteria 3**, it should be noted that a region R is convex if and only if for arbitrary two points $\mathrm{x}, \mathrm{y} \in \mathrm{R}$, the line segment $\overline{xy}$ is inside R.

Hence, since A and B are both convex, for $\mathrm{x}, \mathrm{y} \in \mathrm{A}$ and $\mathrm{x}, \mathrm{y} \in \mathrm{B}$, $\overline{xy} \in \mathrm{A}$ and $\overline{xy} \in \mathrm{B}$ correspondingly. Therefore, $\overline{xy} \in \text{intersetion of A and B, i.e. C.}$

By definition of convex hull, because it is the intersection of all convex regions containing a finite point set, convex hull is convex.

$\mathrm{Q.E.D}$

Linear Combination/Convex Combination

In linear algebra, the linear combination is an expression written in the dot product of a series of position vectors and their corresponding constants in a given range, i.e. $a_1\overrightarrow{p_1} + a_1\overrightarrow{p_2} + \cdots a_1\overrightarrow{p_n}$ or $\begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} \cdot \begin{pmatrix} \vec{p}_1 \\ \vec{p}_2 \\ \vdots \\ \vec{p}_n \end{pmatrix}$. Notably, different linear combinations can specify particular regions or spaces in geometry. For example, with all constants $a_i \in \mathbb{R}$, the linear combination can well define a *d-dimensional* space, in which *d* is the number of independent vectors in the combination. Here, however, we only focus on **regions** defined by linear combinations, where $\overrightarrow{p_i} = \begin{pmatrix} p_i^1 \\ p_i^2 \end{pmatrix}$, and investigate if they can specifically represent a convex hull by the given point set (since points can also be considered as position vectors).

Linear Combination of two points

First, we'll start with two points (i.e. position vectors) on the plane and explore if we can express all points on line AB in position vector form. The investigation is shown below:

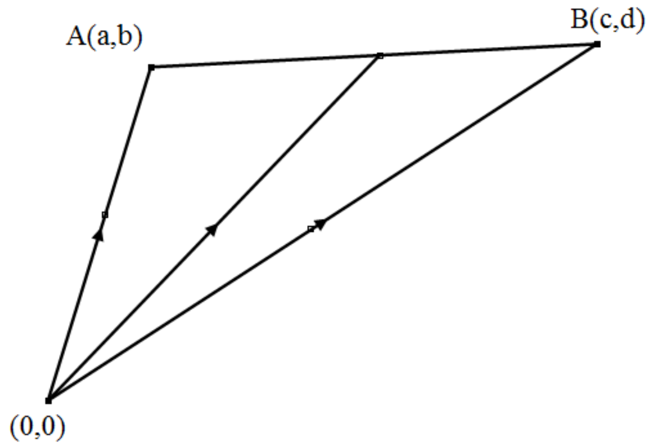

Fig 5. Linear Combination of two points

<u>Investigation:</u> Consider origin $(0,0)$, point $A(a, b)$ and $B(c, d)$ on the plane. Denote position vectors of A (i.e. $\binom{a}{b}$) and B (i.e. $\binom{c}{d}$) as $\vec{x}$ and $\vec{y}$, respectively. Hence, the vector from A to B is $\vec{y} - \vec{x}$ (i.e. $\binom{c-a}{d-b}$). Consider a constant $\lambda \in [0,1]$. Thus, any point on line $\overline{AB}$ can be expressed as $\vec{x} + (1-\lambda)(\vec{y} - \vec{x})$ by vector addition. Arrange the expression as follows:

$$\vec{x} + (1-\lambda)(\vec{y} - \vec{x}) = \binom{a}{b} + (1-\lambda)\binom{c-a}{d-b} = (1-\lambda)\vec{y} + \lambda\vec{x}$$

Hence, it could be concluded that all points on line $\overline{AB}$ can be expressed with linear combinations of $\vec{x}$ and $\vec{y}$. Furthermore, if we add another point, C, on the plane, by mathematical induction, we can also express all lines connecting C to $\overline{AB}$ with linear combinations and thus, express the whole triangle $\Delta\mathrm{ABC}$ (i.e. the convex region is defined by vertices $\mathrm{A, B,}$ and C).

Linear Combination of convex hull

From our investigation results, similarly, we can make a conjecture about convex hull as follows:

Convex Combination

Claim: *Given a finite point set in* $\mathbb{R}^2$**,** $S = \{p_1, p_2, \dots p_n\}$,

$conv(S) = \{\vec{v} \mid a_1\overrightarrow{p_1} + a_2\overrightarrow{p_2} + \cdots a_n\overrightarrow{p_n}\}$ *with all* $a_i \geq 0$ *and* $\sum_{i=1}^{n} a_i = 1$. *(i.e. convex hull can be expressed by a linear combination with all position vectors of the given finite point set in* $\mathbb{R}^2$.*)*

Proof : The proof develops an induction from the proof of two-point linear combination and the complete proof is presented in the appendix[1].

Implication

By geometric significance of linear combination (i.e. linear span), we can simply apply its features to convex hull and hence, help us to develop our algorithms in next section. Here're two key properties below:

(1) The trajectory (outer-bound) of a convex hull is defined by a subset of points on the plane as **vertices**.

(2) The vertices of a convex hull are **linear interpolation** and therefore, convex hull displays as a convex polygon shape.

Convex Hull's Algorithm

Applying convex hull's properties we just discussed above, in this section, we will particularly take an insight into efficient approaches to constructing convex hull in $\mathbb{R}^2$ with further comparison tests. To start with, we'll first explore some general ideas about the essence (i.e. basic strategy) of our algorithms. Based on these discoveries, we'll then develop our approaches. What's more, for each approach, we'll have an approach outline and a detailed algorithm description. Some neat proofs will be given after each algorithm.

Problem Definition

[1] See Appendix A

Given a finite point set $S = \{p1, p2, p3 \dots, pn\}$ $(n \geq 3)$ in $\mathbb{R}^2$, find the convex hull of S, conv(S), with most efficiency.

Basic Strategy

By implications that convex hull is, in fact, a minimal convex polygon defined by the subset V (i.e. vertex set) of the given points $\{p_1, p_2, \dots p_n\}$, we can try starting by some (or even singe) points in the set and considering how to connect them (it) to some others. Furthermore, it should be noted that points with the most and least x-coordinates (i.e. the leftmost, rightmost points) as well as points with the most and least y-coordinates (i.e. the topmost and bottommost points), what here we call extreme points, are necessarily included in the vertex set V using proof by contradiction. Hence, our strategy follows that we **start by some extreme points on the plane and try connecting some refined (known) vertices to other points in each step**.

Approach Outline (Convex-Merge)

Given a convex hull of (n $-$ 1) points on the plane, i.e. $\{p_1, p_2, p_3 \dots, p_{n-1}\}$, now adding another point $p_n$outside it, we want to consider if p_n can be merged into the existing convex hull so that the new region is the convex hull of the entire n points. This is especially useful as it follows the structure of an algorithm so that we **can start from a convex hull of the least points (i.e. three for it's the least number of points required to form a convex hull) and merge other points in the set by a certain order**. In addition, noticing that convex hull is actually a convex polygon defined by points in the given point set, when we merge the next point, we shall **only consider connecting that point to some of the existing convex hull's vertices**. To arrange the algorithm into a nice order, we can **merge points from the top to the bottom** so that we won't miss any of them.

Algorithm

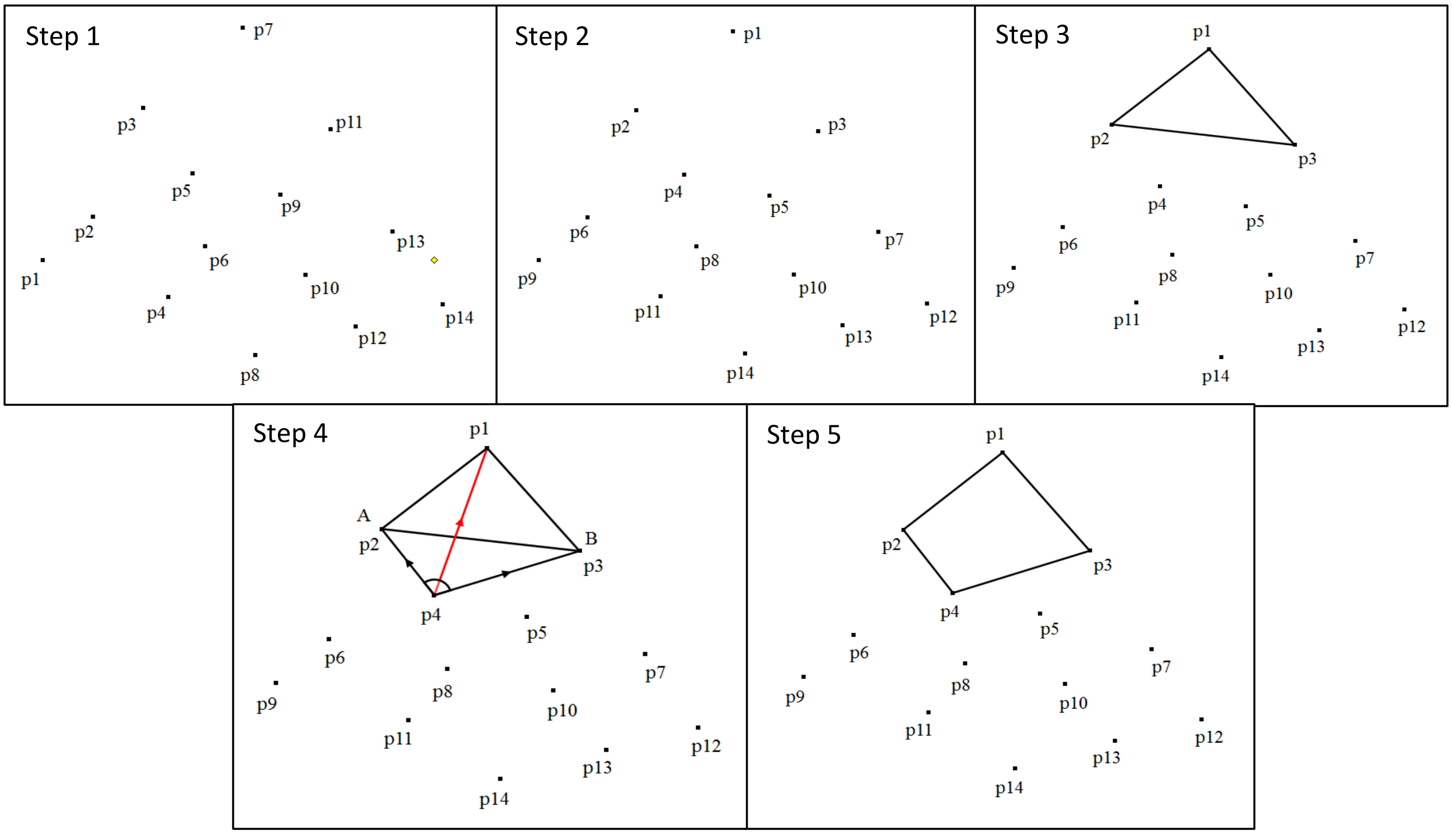

Fig 6. Applying Convex-Merge algorithm to add p_4 into convex hull

Step 1. Input the finite point set $S = \{p1, p2, p3 \ldots, pn\}$ in $\mathbb{R}^2$

Step 2. Sort all points with respect to their y-coordinates in descending order into a new set Q

Step 3. Take the first three points in Q and connect them to form a triangle

Step 4. Take the next point p_i remaining in Q and connect it to all vertices of the refined polygon it can "see". Note the two segments that form the greatest convex angle at p_i as Ap_i and Bp_i, where A is on the left of B.

Step 5. Refine the new polygon by removing all segments between Ap_i and Bp_i.

Step 6. Repeat from 3) to 5) till there's no point remaining in Q. The resultant region is the convex hull, conv(S), of S.

Proof

Problem Restatement: The Convex-Merge algorithm is valid for any finite point set $S = \{p_1, p_2, \dots p_n\}$ $(n \geq 3)$ in $\mathbb{R}^2$ to construct the convex hull, conv(S).

Proof by Induction:

Base Case: When $n = 3$, the algorithm will process to step 3) and directly skip to 6). At the time, there's only a triangle connecting all three points on the plane. As we know convex hull is the minimal convex polygon defined by the subset of the point set, in this case, the convex hull conv(S), is obviously a triangle connected all three points.

Q. E. D.

Inductive Hypothesis: Assume the Convex-Merge algorithm is valid for the finite point $S = \{p_1, p_2, \dots p_k\}$ in $\mathbb{R}^2$ to construct the convex hull, conv(S).

Prove: the Convex-Merge algorithm is valid for the finite point set $S = \{p_1, p_2, \dots p_{k+1}\}$ in $\mathbb{R}^2$ to construct the convex hull, conv(S).

Inductive Proof: Apply the algorithm on the finite point set $S = \{p_1, p_2, \dots p_{k+1}\}$ in $\mathbb{R}^2$. Consider the algorithm is processed till there's only one point remaining in sorted point set Q. Note the point as p_i. By inductive hypothesis, now we have a convex hull of k points on the plane. Merge the last point to the existing convex hull so that a new region is formed. Notably, the **k-point convex hull is a subset of the new region** as fundamentally, we just add a region to it by two intersecting segments. To prove the algorithm is valid, we need to prove both that **the new region is convex** and **the convex**

region is minimal to enclose all $(\mathbf{k}+\mathbf{1})$ points.

1. **The new region is convex (Proof by contradiction)**

Assume the region is concave. The remaining angles of the k-point convex hull in the new region are definitely all convex. By the description of the algorithm, $\angle p_i$ is convex as well. Therefore, the concave angle can only occur at ∠A or ∠B.

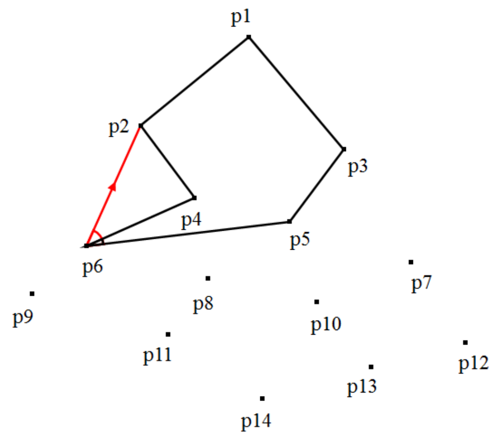

Fig 7. Proof of convexity

Consider ∠A is concave. By criteria 1 of convexity, when the "guard" is at p_i, he mustn't be able to "see" all points in the region. Furthermore, it should be noticed that he will also "see" another vertex C on the left of A. By the description of the algorithm, since p_i can "see" C and $\angle Cp_iB > \angle Ap_iB$, Cp_i should be connected while Ap_i should be removed — This is a great contradiction. Therefore, the new region is convex.

2. **The convex region is minimal to enclose all $(\mathbf{k}+\mathbf{1})$ points**.

As the k-point convex hull is a subset of the new region, the original k points must all be enclosed. What's more, the new point p_i is a vertex of the region. Hence, the convex region encloses all $(\mathrm{k}+1)$ points.

At the same time, as p_i is bottommost in the $(\mathrm{k}+1)$ points, it must be a vertex of the (k+1)-point convex hull. Since the k-point convex hull is the minimal convex region enclosing the original k points, we can prove/disprove the statement just by determining if the added region is minimal. Notably, (with proof by contradiction,) only when we connect to two "reachable" (i.e. p_i can "see") vertices of the original k-point convex hull so that the segments form the largest convex angle at p_i, can we ensure the original k-point convex hull is thoroughly included in the new region (i.e. the new region includes all original k points). Hence, the added region provided by the algorithm is minimal and thus the new region is a minimal convex region enclosiing all $(\mathrm{k}+1)$ points.

As I mentioned above, since the new region is the minimal convex region that encloses all $(\mathrm{k}+1)$ points, it's the convex hull of all $(\mathrm{k}+1)$ points on the plane and thus, the algorithm stands for $(\mathrm{k}+1)$ points.

$\mathrm{Q.E.D.}$

By the principle of mathematical induction, Convex-Merge algorithm is valid for any finite point set $\mathrm{S}=\{p_1, p_2, \ldots p_k\}$ in $\mathbb{R}^2$ to construct the convex hull, conv(S).

$\mathrm{Q.E.D.}$

Algorithm Evaluation

Adopting "Big O" notation — $\mathrm{O}(\text{"polynomial of n"})$ — to express the lower bound of the algorithm's computational complexity, the cost is $\mathrm{O}(n^2)$and the complete analysis is in the appendix[2].

Conclusion & Reflection

To conclude, in this paper, we took a deep insight into several fundamental properties of convex hull of a finite point set in $\mathbb{R}^2$ and proposed a practical approach to computing it with decent efficiency.

Since it's widely considered that the computation of convex hull is a fundamental problem of computational geometry with hard complexity, the algorithm proposed in this paper offers us a quite acceptable baseline and hopefully gives some inspirations for

[2] See Appendix B

future work. Some strengths of the algorithm are discussed following. First, the algorithm can basically be applied to any configuration of the point set on the plane and thus solve the problem without the loss of generality. Moreover, it's even simple for people to follow on their own to find the convex hull of a given point set in $\mathbb{R}^2$. However, there's also some critical weakness people should be aware of. In the algorithm description, we adopt an ambiguous word "see" to indicate the prerequisite for the vertices that we would consider connecting to. While it's intuitive for us to point out what "see" means in the context, for machine, it's not a well-defined concept. Hence, further works are needed to refine the description when implementing the algorithm and it may add a considerable cost to its actual computational complexity.

Besides, notably, our approach is primarily based on the implication that convex hull is the convex polygon defined by the subset of the finite point set. Therefore, it's not a direct use of the link between convex hull and linear combination. In fact, it's intriguing to see that convex hull can be expressed by just a series of equation in terms of the coordinates of the given points on the plane. Hence, finding a purely algebraic approach to solve the problem can be feasible and appealing in future as it would enable the machine to do computations directly based on numbers instead of visualizing the whole geometry, and thus largely improve the efficiency.

Again, it should be emphasized that the work of computing convex hull with high efficiency will boost the development of various fields such as computer vision, GIS system and so on. Although the computational complexity of the algorithm proposed in this paper is incomparable to the one of output-sensitive algorithm (the quickest approach to this problem), we hope to hint more mathematicians to solve the problem.

References

[1] Weisstein, Eric W. "Convex Hull." From MathWorld--A Wolfram Web Resource.

http://mathworld.wolfram.com/ConvexHull.html

Appendix A – Proof of convex combination

***Claim**: Given a finite point set in* $\mathbb{R}^2$**,** $S = \{p_1, p_2, \dots p_n\}$,

$conv(S) = \{\vec{v} \mid a_1\overrightarrow{p_1} + a_2\overrightarrow{p_2} + \cdots a_n\overrightarrow{p_n}\}$ *with all* $a_i \geq 0$ *and* $\sum_{i=1}^{n} a_i = 1$. *(i.e. convex hull can be expressed by a linear combination with all position vectors of the given finite point set in* $\mathbb{R}^2$*.)*

Proof: Denote the convex hull and the linear combinations of the given finite point set, S, as conv(S) and M respectively. To prove the claim is true, we need to prove in both directions — $\mathbf{M} \sqsubseteq \mathbf{conv(S)}$ and $\mathbf{M} \sqsupseteq \mathbf{conv(S)}$.

1. $\mathbf{M} \sqsubseteq \mathbf{conv(S)}$ (**Proof by induction**)

Here, we want to apply induction on n for $S = \{p_1, p_2, \dots p_n\}$and prove all linear combinations of $\{p_1, p_2, \dots p_i\}$ are in conv(S).

Base Case: Let $\mathrm{i} = 1$ and $\mathrm{S} = \{p_1\}$. Hence, $\mathrm{M} = \{p_1\} = S \sqsubseteq \mathrm{conv(S)}$ $\quad$ Q.E.D.

Inductive Hypothesis: Assume $p_1, p_2, \dots p_i \in S$ and all their linear combinations $a_1\overrightarrow{p_1} + a_2\overrightarrow{p_2} + \cdots a_n\overrightarrow{p_i}$ are in conv(S).

Prove: for $p_1, p_2, \dots p_{i+1} \in S$, their linear combinations $a_1\overrightarrow{p_1} + a_2\overrightarrow{p_2} + \cdots a_n\overrightarrow{p_{i+1}}$ are also conv(S).

Inductive Proof: Arrange the linear combinations of $p_1, p_2, \dots p_{i+1}$ as follows:

$$a_1\overrightarrow{p_1} + a_2\overrightarrow{p_2} + \cdots a_n\overrightarrow{p_{i+1}} = (a_1 + a_2)\left(\frac{a_1}{a_1 + a_2}\overrightarrow{p_1} + \frac{a_2}{a_1 + a_2}\overrightarrow{p_2}\right) + a_3\overrightarrow{p_3} + \cdots a_n\overrightarrow{p_{i+1}}$$

By inductive hypothesis, it can be infer that $\frac{a_1}{a_1+a_2}\overrightarrow{p_1} + \frac{a_2}{a_1+a_2}\overrightarrow{p_2}$ is another point $\in$ conv(S). Also, since $\underbrace{\frac{a_1}{a_1+a_2}\overrightarrow{p_1} + \frac{a_2}{a_1+a_2}\overrightarrow{p_2},\ \overrightarrow{p_3} \dots \overrightarrow{p_{i+1}}}_{i\ points} \in conv(S)$, the linear combinations of these i points are all in $\mathrm{conv}(S)$ (i.e. the linear combinations of $p_1, p_2, \dots p_{i+1}$ are all in $\mathrm{conv}(S)$). $\qquad \mathrm{Q.E.D.}$

By the principle of mathematical induction, all linear combinations of $\{p_1, p_2, \dots p_i\}$ are in $\mathrm{conv}(S)$. (i.e. $\mathrm{M} \sqsubseteq \mathrm{conv}(S)$) $\qquad \mathrm{Q.E.D.}$

$2.\ \mathbf{M} \sqsupseteq \mathbf{conv}(\mathbf{S})$

By definition of linear combination, $\mathrm{S} \sqsubseteq \mathrm{M}$. Therefore, we want to show that M is convex (so that M is a convex region containing $\mathrm{S} \Rightarrow \mathrm{M} \sqsupseteq \mathrm{conv}(S)$).

Consider two arbitrary points $\mathrm{x}, \mathrm{y} \in \mathrm{M}$, where $\begin{cases} \vec{x} = a_1\overrightarrow{p_1} + a_2\overrightarrow{p_2} + \cdots a_n\overrightarrow{p_n} \\ \vec{y} = b_1\overrightarrow{p_1} + b_2\overrightarrow{p_2} + \cdots b_n\overrightarrow{p_n} \end{cases}$. By the results of our investigation, we can express all points, z, on line $\overline{xy}$ as follows:

$$\mathrm{z} = \lambda\vec{x} + (1-\lambda)\vec{y} = \sum_{i=1}^{n} (\lambda a_i + (1-\lambda)b_i)p_i$$

Thus, line $\overline{xy}$ is also in M. Furthermore, by “Convex region criteria”, M is convex and therefore, $\mathrm{M} \sqsupseteq \mathrm{conv}(S)$. $\qquad \mathrm{Q.E.D.}$

Since $\mathrm{M} \sqsubseteq \mathrm{conv}(S)$ and $\mathrm{M} \sqsupseteq \mathrm{conv}(S)$, $\mathrm{conv}(S) = M$.

$\mathrm{Q.E.D.}$

Appendix B – Algorithm Evaluation

For evaluation, we would consider the worst case of the algorithm and take it as the lower bound of our approach. Also, we would adopt "Big O" notation — $\mathrm{O}(\text{"polynomial of n"})$ — to express the computational complexity of an algorithm. It indicates that the performance of the algorithm won't be below the polynomial in terms of the size of the finite point set, n.

Consider there're n points on the plane. For the sorting step (i.e. step 2), we would simply apply a most efficient algorithm known so far to all points (*divide-and-conquer algorithm*) and it will give us a lower bound of $\mathrm{O}(n^2)$. In step 3, it would take constant time to connect the three points as there's no order of growth by the size of the point set.

Consider step 4 with p_i as the next point merged to the convex hull. In this case, the algorithm will check all vertices of the previous region to determine if p_i can "see" any of them. In the worst case, the vertices will always be the most, i.e. $(\mathrm{i}-1)$. Hence, the computation cost would be a $\sum_{i=3}^{n} i = \mathrm{O}(n^2)$ (where a is a constant). Meanwhile, when checking all those vertices p_j one after another, we can also note their positions in terms of the gradient of $p_i p_j$. Therefore, while determining the largest angle formed at p_i, it wouldn't add any cost on computational complexity at all.

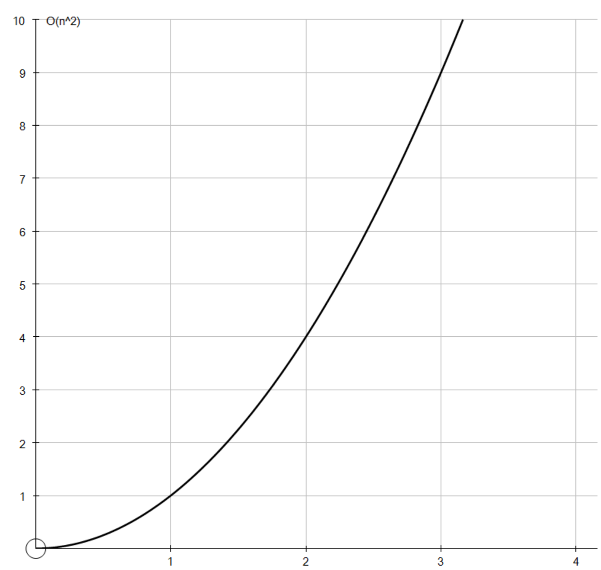

Fig 8. Order of growth of $\mathrm{O}(n^2)$

For final clean-up in step 5, there're at most i edges in a convex hull (convex polygon).

Thus, in the worst case, we'll remove all i edges for the newly added point p_i(though it turns out to be impossible) and spend a $\sum_{i=3}^{n} i = \text{O}(n^2)$ (where a is a constant) in total. Overall, by summing up all the costs spent by each step of the algorithm — $\text{O}(n^2) + \text{O}(1) + \text{O}(n^2) + \text{O}(n^2) = \text{O}(n^2)$ — the lower bound of Convex-Merge algorithm is $\text{O}(n^2)$.