Optimisation stochastique

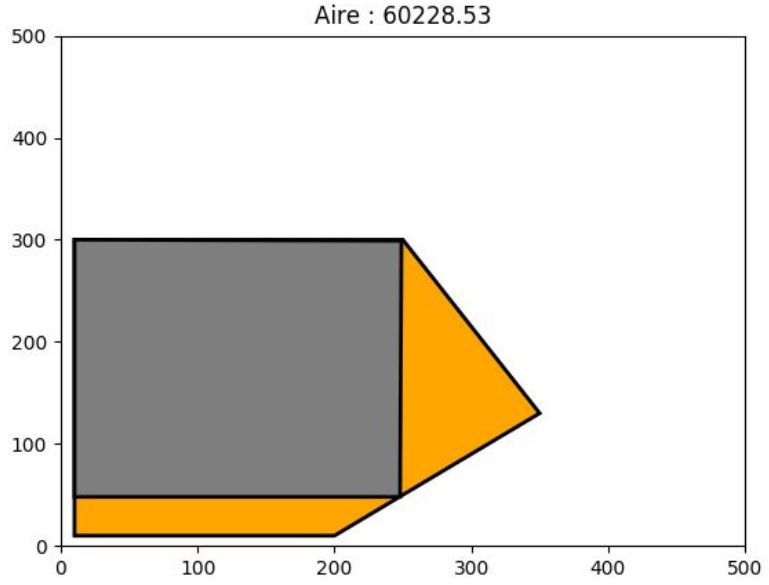
Tp : Maximisation d'une surface constructible dans une parcelle.

EISTI
Cergy - Pau

La problématique

Un cabinet d'architecte pose le problème (simplifié) suivant : "*Dans une parcelle constructible, trouver l'emprise au sol du bâtiment de plus grande surface, contenu dans la parcelle*".

- Étant donné un polygone quelconque (convexe ou concave), le but est de trouver le plus grand rectangle contenu dans celui-ci.



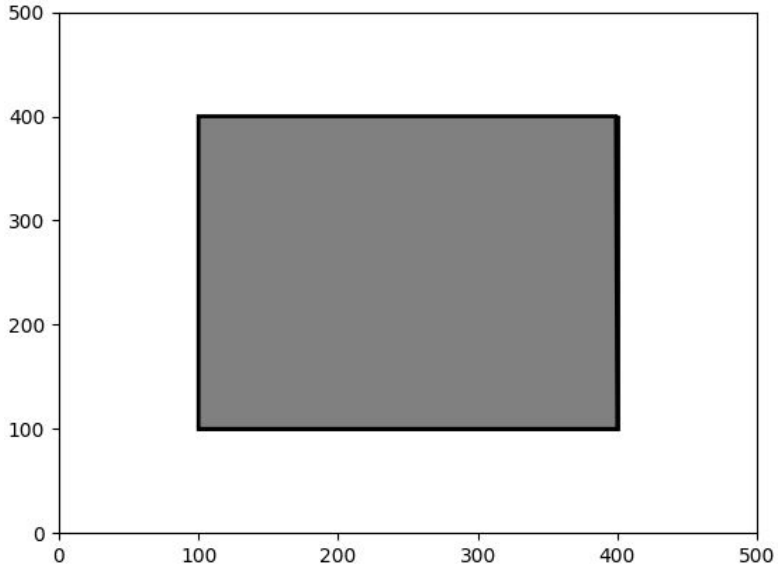
Le problème d'optimisation

Les composantes du problème sont :

- Le polygone : l'espace de recherche contraint ;
- Le rectangle : la solution du problème ;
- La faisabilité (le rectangle est-il inscrit dans le polygone ?) : un problème contraint ;
- L'aire du rectangle : la fonction d'évaluation ;
- => problème de *maximisation*.
- **PRINCIPALE DIFFICULTÉ : DÉCRIRE LE PROBLÈME**

Le polygone

Un polygone est un n-uplet de couples représentant les coordonnées (abscisse,ordonnée) de chaque sommet :



polygone = ((100,100),(100,400),(400,400),(400,100))

Le rectangle

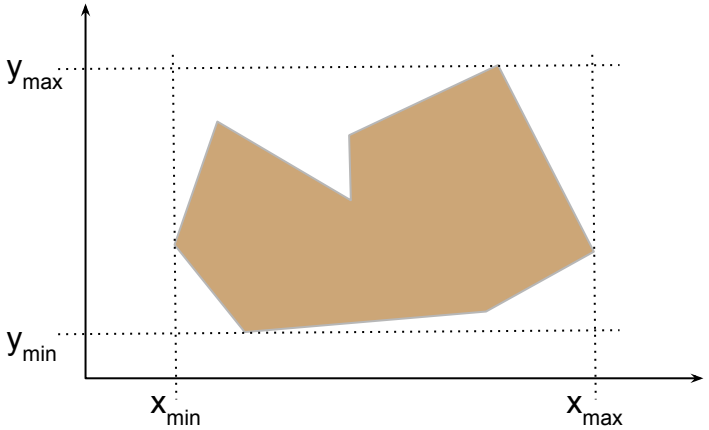
Comment modéliser un rectangle ?

- Limiter le nombre de paramètres (réduire la dimension du problème) ;
- “Penser voisinage” : être certain que le “voisin” d’un rectangle est un rectangle ;
- Comment choisir sa représentation afin de parcourir efficacement l’espace de recherche ?

L’espace de recherche

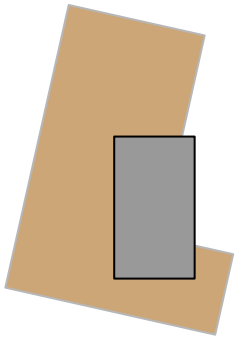
Définir l’espace de recherche des coordonnées du rectangle selon le polygone.

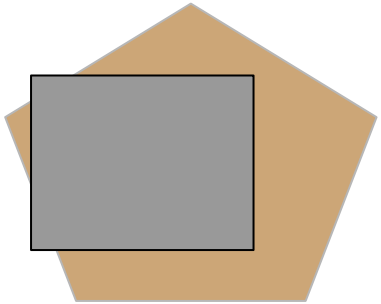
- Boîte englobante du polygone :

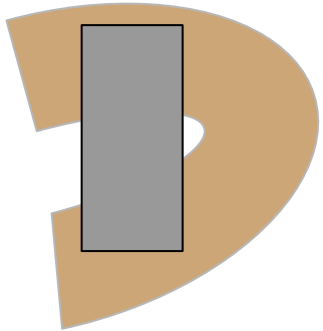


La faisabilité

Un rectangle est-il valide pour le problème ?







Algorithmes de clipping :

- Vatti, Weiler-Atherton, Greiner-Hormann, Sutherland-Hodgman...

To Do List - 1/2

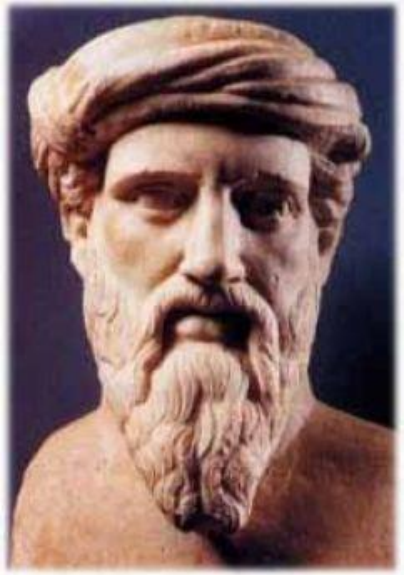
Dans un premier temps, on vous demande de décrire le problème :

- Modéliser un rectangle comme solution candidate du problème ;
- Écrire la fonction `sol2rect(solution)` qui transforme une solution du problème en rectangle (n-uplet de coordonnées) ;
- Tester la libraire pyclipper, permettant de faire du clipping selon l'algorithme de Vatti. Comprendre son fonctionnement et écrire un prédicat `estValide(polygone,rectangle)` qui vérifie que le rectangle est bien contenu dans le polygone ;
- Écrire la fonction objectif à maximiser : `aire(rectangle)`.

To Do List - 2/2

Dans un second temps, on vous demande de solutionner le problème d'optimisation et de réaliser des tests statistiques sur leurs performances :

- Adapter et appliquer 2 algorithmes parmi ceux abordés en cours ;
- Définir un critère de comparaison équitable ;
- Créer un échantillon de 30 résultats par algorithme ;
- Faire les boîtes de Tuckey correspondant aux résultats sur un même graphique ;
- Utiliser un test statistique **approprié** pour comparer les algorithmes entre eux.



That’s all folks !