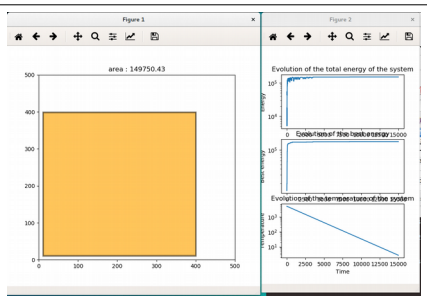
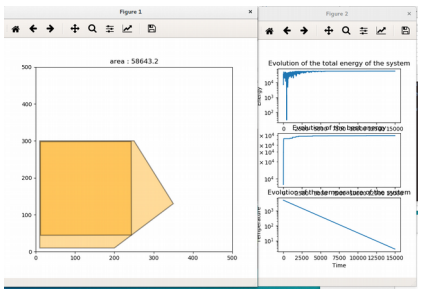
Name: LE Thu Huong
Class: ADEO1

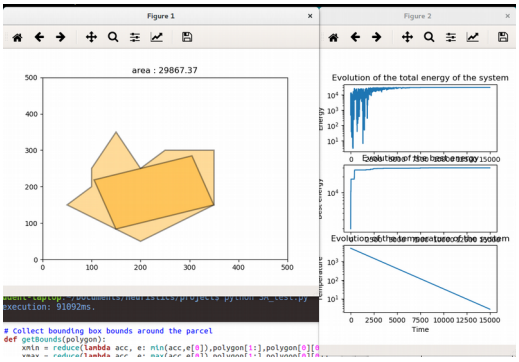
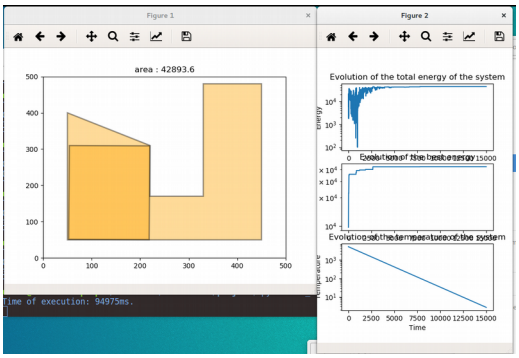
FINAL PROJECT REPORT FOR HEURISTIC

PROBLEM: Maximization the rectangle inside a polygon

1. Ideal explanation:
getBounds() to get the boundary of a polygon: return xmin, xmax, ymin, ymax
pos2rect() to get the point a,b,c,d of rectangle : return a,b,c,d
*However, function require to rotate the rect to generate all possible solution for SA algr()
* Main math to apply: please watch this great video (15min) of Khan academy
https://www.khanacademy.org/math/linear-algebra/matrix-transformations/lin-trans-examples/v/linear-transformation-examples-rotations-in-r2
distance() return euclidean distance
area() return area of rectangle
verifConstraint() return a sharp (return true to 4 point, others are return false)
* install clipper in Linux $ pip install clipper
* https://pypi.org/project/pyclipper/
initOne() generate a random solution
Other functions are realization of existing algorithms

2. Tried algorithm : Simulated annealing and Particle swarm
3. Compare results of the same problem set
* Time complexity : Simulated Annealling : O(n2) vs Particle swarm O(n2)
* S-best: by average and standard deviation.
Because of TSP is a stochastic problem, I ll take the average of 3 S-best (best solution) provided by simulated annealling and Tabu search. Then compute the standard deviation for each number of cities.

Problem size	Simulated Anealling result T0 = 5000; IterMax = 150000	Tabu search result Nb_cycles = 500 Nb_particle = 20
1st polygon	 Mean: 149134 Std = 638.17	Psi = 0.3; Cmax = 1.3 Mean:92141 Std: 28249.85 results are very bad so we change other params
2nd polygon	 	Psi = 0.7 ; Cmax = 1.7 Mean:58139 Std:561.3

	Mean: 58722.3 Std = 366.6	
3rd polygon	 Mean:29729 Std: 125.677	Psi = 0.7 ; Cmax = 1.7 Mean: 29256.67 Std: 1047.69
4th polygon	 Mean : 43982.6 Std: 1512.122	Psi = 0.7 ; Cmax = 1.7 Psi = 0.9; Cmax = 1.9 → very bad Mean: 35504.3 Std: 4814.5