OPTEC Labs : Optimization of a trajectory AB inside a 10x10 room by avoiding collisions against obstacles using MATLAB

Author:
Thomas MARQUET
Leonid LIANOU

Supervisor:
Fouad Bennis

Monday 20th May, 2019

1 Introduction

The purpose of this lab is to understand how to use the function fmincon and the methodology around it. But also to know the limits of this tool.

To implement the obstacles we choosed to use fixed starting point, end point and circles because it is more consistent. The optimized path is in red and the guessed trajectory is in green. Comments about the code on himself

are in the code.(discretization inside contraint4, fmincon and its arguments inside project-main-final, and objective function in obj4).

2 First step : 1 circle and 1 guess point

As you can see below it is really simple :

We observed that the optimized path could be either going over the circle or under the circle. It is a simple phenomenon of local minimum depending on the guess points. If the guess points are mostly over the circle, then the final path would be over the circle even if going down would be even more optimized.

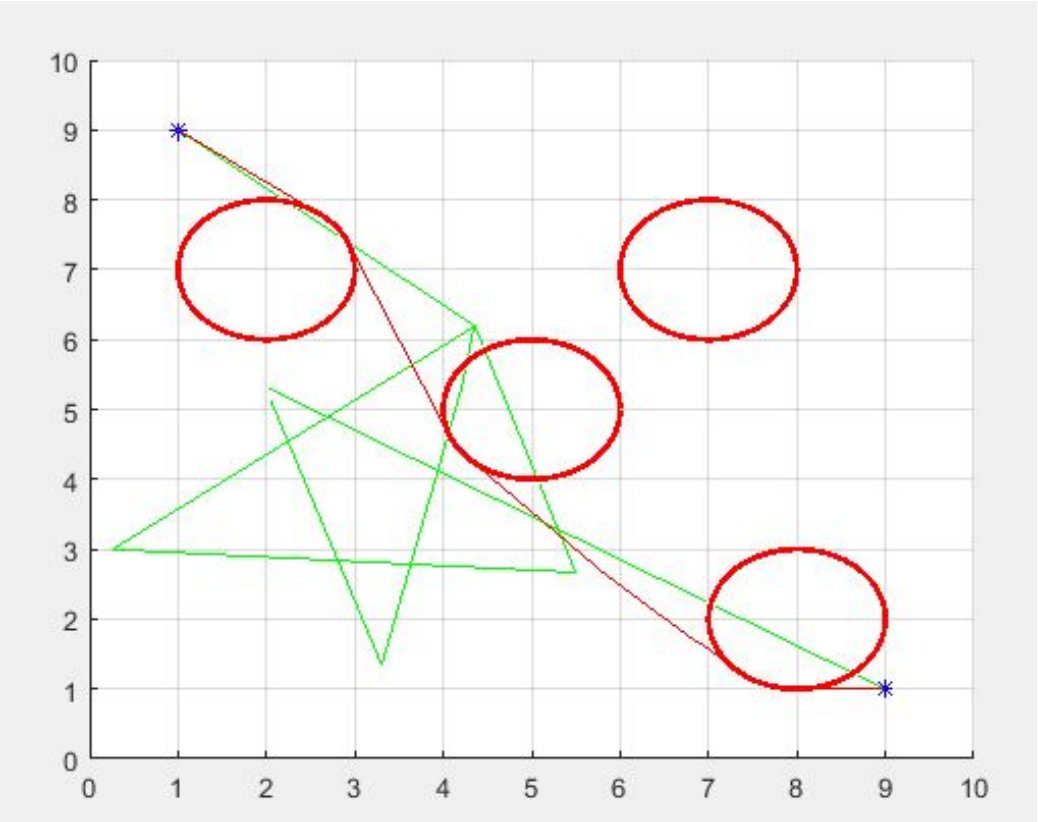
3 Second step : 2 circles and 5 guess points

For this step, we've got five guess points so because of the weird shape of the initial trajectory and the randomness of the guess points, it gets less consistent. points.

Same as before, the optimized path highly depends on where are the initials points but in this case convergence of the path is not systematic also because of the guess points.

4 Third step : 4 circles and 7 guess points

Just like the second step, the more we add obstacles and randomness into the problem, the less it is consistent.

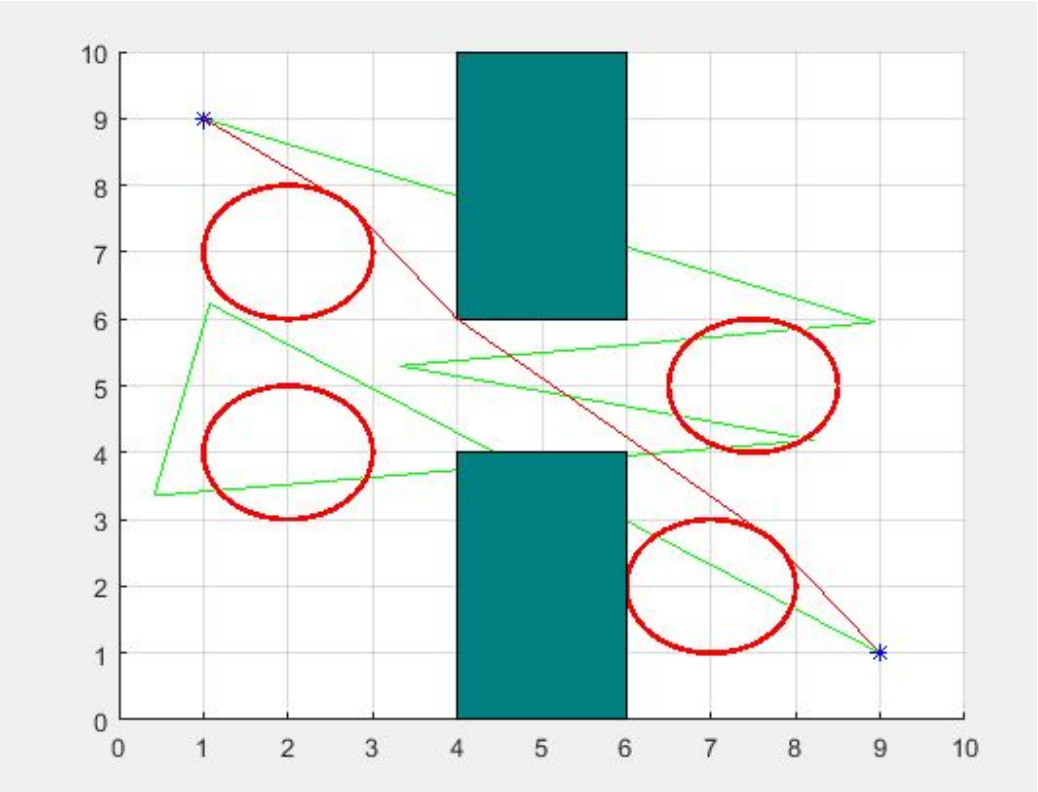


Same as before, the optimized path highly depends on where are the initials points.

5 Final step : 4 circles and walls

For the final step, the methodology is quite different since we now have walls and it is adding a lot more difficulties to converge. So we've to help the algorithm by fixing the right boundaries to the initials guess points. We can have, for example, on 7 guess points : 2 guess points on the left of the walls, 2 on the right, 1 at the entry of the corridor, 1 in the middle of the corridor and 1 at the end.

The ”head start” we’re giving to the algorithm is a critical condition on its convergence.



Same as before, the optimized path highly depends on where are the initials points. But since here the guess points are less ”free”, the path is most likely to be similar on each try.

6 Conclusion :

We can see that in every case the path depends on the guess points. With two differents sets of guess points we can have drastically differents solutions and even solutions that would never have been taken by an human (really far from global minimum). I guess a good solution to improve this would be to take multiple convergent paths and compare them to get the best answer between all of them. Or you could also get your initial guess point using known acceptable guess points(like in the final solution).