Path planning by hybrid PSO-Splines algorithm

Paulo Salgado

CITAB/ECT-Departamento de Engenharias, Universidade de Trás-os-Montes e Alto Douro, Vila Real, Portugal

Introduction

This paper presents a new version of the Particle Swarm Optimization algorithm where the particles are replaced by spline functions. The developed algorithm generates smooth motion trajectories with two times continuously differentiable curvature avoiding obstacles placed in the workspace. It can be used for autonomous robot path planning or transport problems. The spline based trajectory generation gives us continuous, smooth and optimized path trajectories. Simulation and experimental results demonstrate the effectiveness of the proposed method.

PCO-CS Algorithm

The PSO-CS use splines *S* instead of particles *P*. However, for simplicity of analysis, we are going to use here the same designation indistinctly. S_i represents a Spline with waypoints X_i. This spline belong to W space and has instantaneous velocity vector $\vec{u}_i$, which is a Cubic Spline interpolation vector function. It has velocity waypoints at $\vec{U}_{ij} = \vec{u}_i(t_j)$ for the positional waypoint X_{ij} at instant t_j. As other PSO algorithms, it remembers its individual best value of fitness function and position $\bar{S}_i$ as well as the global best $\bar{S}_G$ of the whole swarm, stored in . During each iteration, the velocity update rule:

$$\vec{u}_i^{(k+1)}(t) = \alpha \vec{u}_i^{(k)}(t) + C_1 r_1\left(\bar{S}_i(t) - S_i(t)\right) + C_2 r_2\left(\bar{S}_G(t) - S_i(t)\right) + C_3 r_3 \vec{R}_i(t) \qquad S_{i+1}^{(k+1)}(t) = S_i^{(k+1)}(t) + \vec{u}_i^{(k+1)}(t)$$

The repulsion force, $\vec{R}_i(t) = \bar{S}_i(t) - CO_o,\ \text{for } t' < t < t''\text{ ; 0 otherwise}$, acts on the Spline i, or in the portion that violates the space of the object, result of cumulative effect of all objects. It has a radial direction, from the center of the objects, CO, and equal intensity for length of line.

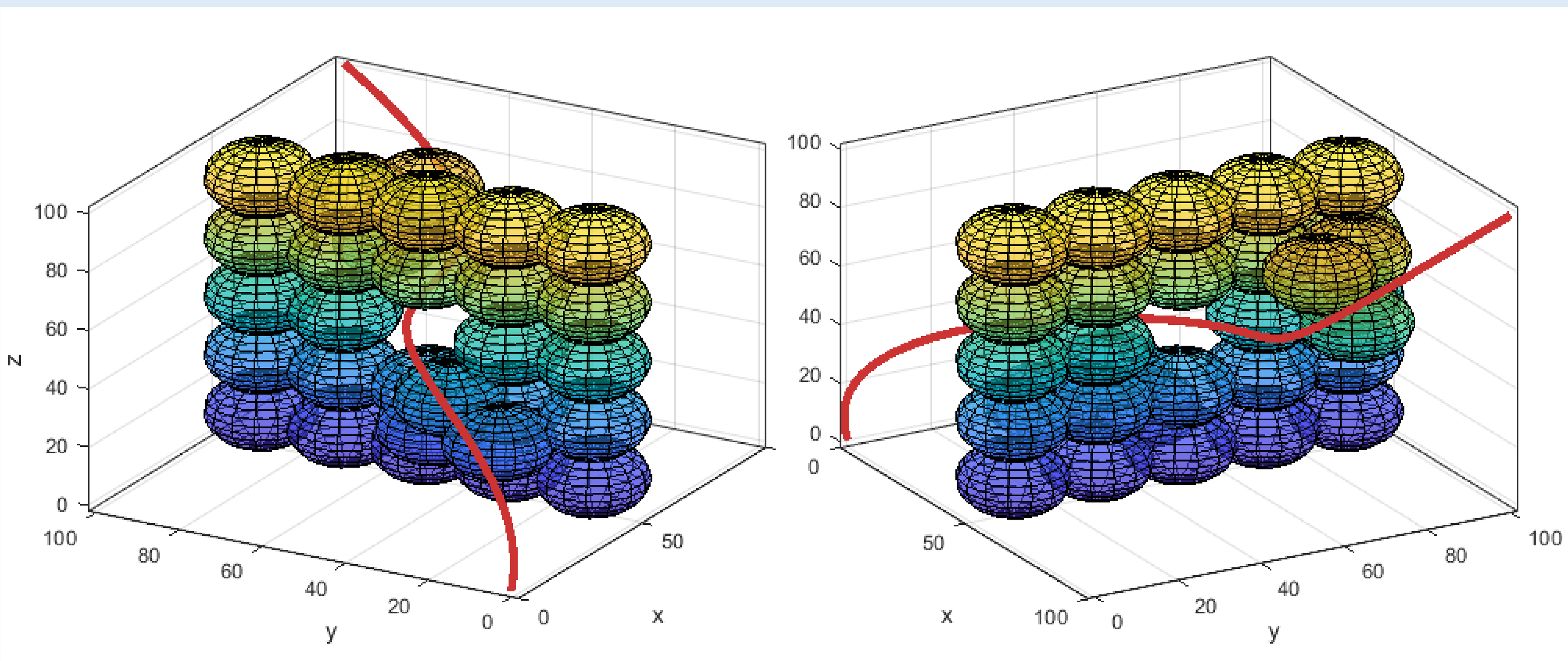
The fitness of the spline is computed by $F_i = nv_i\left(1 + \beta \cdot VA_i\right) + LS_i + \delta \cdot SD_i$ where β and δ are weight parameters, *nv* is the number of object violated by the ith spline, with the violating area *VA*, *LS* is the length of trajectory (Spline) and *SD* a function that measures the safety of trajectory.

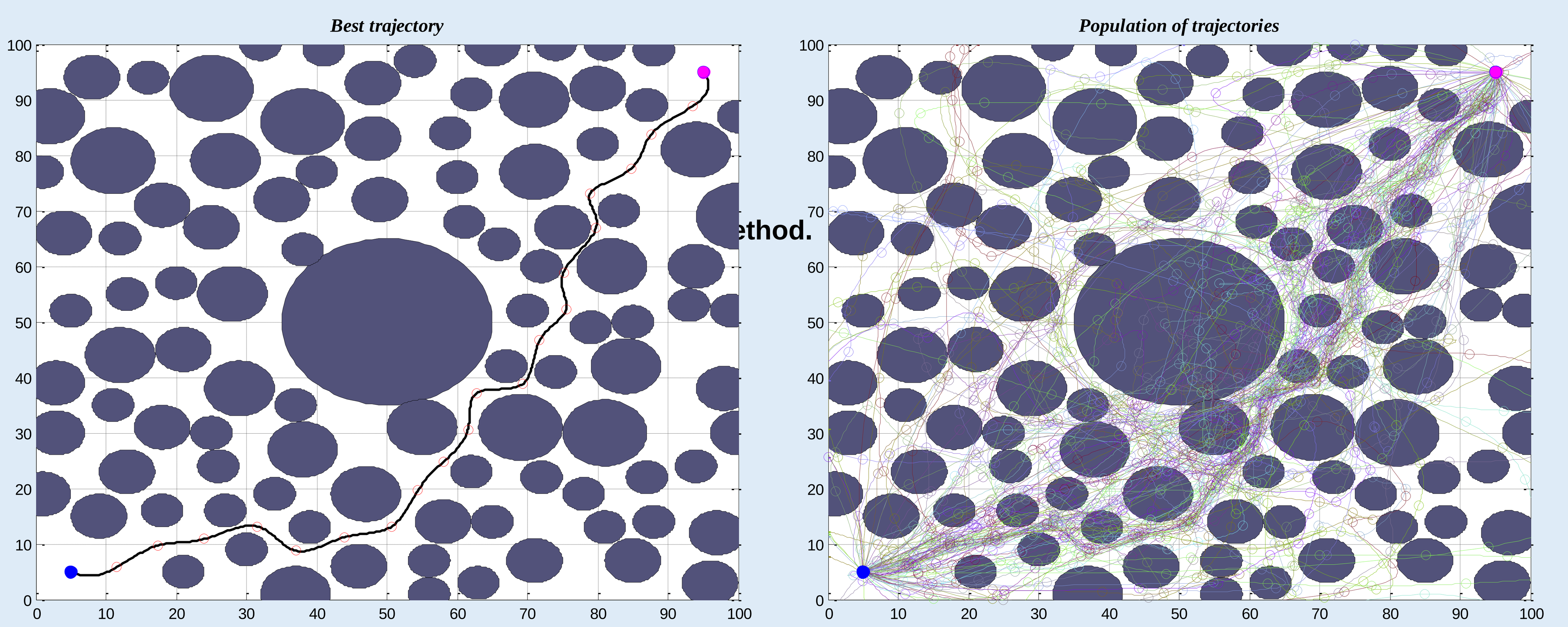
Methodology

PSO-CS algorithm is used for path planning in 2 D and 3D test-examples. The workspace has a shape like a square/cube with edge length of 100 units. Inside there are 30 circular/spherical obstacles with radius length of 10 units. In the 1^{ed} example, the circles are random placed on workspace while in the 2^{ed} text-example a wall, made by 24 spheres and a hole at its centre that split the workspace in two zones. Three more spheres randomly placed in each one-sided zones. The start position of the robot is in origin and the end position is in opposite vertices. It uses Spline curves as particles. One hundred 'particles' have been used for simulations with 100 iterations.

Results

Figures show the results, respectively, for the 2D and 3 D test-examples. The red line shown in figures represents the optimum path generated by the algorithms and the filled circles/spheres represent obstacles.





Conclusion

In this paper, I propose a new PSO-CS for global path planning. In order to get smoother planned paths, the proposed algorithm uses a Cubic-spline smoothing technique. It is testes to robot motion planning problem, which it is treated as an optimization problem. Random obstacles are place on the 2D and 3D workspaces. The PSO-CS, in each iteration, try's to finds a feasible Spline curve with best performance, defined by appropriated waypoints. The result is a smoother planned path, which it is a quasi-optimal trajectory.

Bibliography

- P. Salgado, and at. al, "Hybrid PSO-cubic spline for autonomous robots optimal trajectory planning," 2017 IEEE 21st Int. Conf. on Intelligent Engineering Systems (INES), Larnaca, pp. 131-136, 2017, doi: 10.1109/INES.2017.8118542.
- M. Elbanhawi and at. al., Continuous path smoothing for car-like robots using b-spline curves, J. Int. Rob. Syst., vol. 80(1), pp. 23–56, 2015.