

# INF421 PI: PATH PLANNING ALGORITHMS

Merouane Benadda, Antoine Fèvre

February 13, 2026

**Question 1.** We chose to implement the algorithm in C++, as it allows better performance. For visualization, we used Python with Matplotlib, as it provides a more convenient way to create visual representations of the results.

**Question 2.** See the implementation in the file `scripts/visualize.py`.

**Question 3.** See the implementation in the file `src/pso.cpp`.

**Question 4.** See the implementation in the file `src/pso.cpp`.

**Question 5.**

**Question 6.**

**Question 7.** See the implementation in the file `src/pso.cpp`.

**Question 8.**

**Question 9.**

**Question 10.**

**Question 11.**

**Question 12.** To improve the performance of the algorithm, we could refine the fitness function. Instead of simply being the euclidian distance plus infinity if the path crosses an obstacle, we can make it output the euclidian distance plus a penalty proportional to the distance crossed in the obstacles.

This way, the algorithm would converge faster, especially in cases with lots of obstacles, as it would be able to differentiate between paths that are close to the optimal one but cross an obstacle and paths that are far from the optimal one but do not cross any obstacle.

To do so, we compute the intersection of the path with the obstacles in `src/utils.cpp`. To avoid any situation where the algorithm would prefer crossing an obstacle to reduce the distance, we can set the penalty to be a large constant multiplied by the distance crossed in the obstacle.

**Question 13.**

**Question 14.**

**Question 15.**

**Question 16.**

**Question 17.**

**Question 18.**

**Question 19.**

**Question 20.**

**Question 21.**

**Question 22.**

**Question 23.**

**Question 24.**

**Question 25.**