1-Our state is comprised of two lists. The first two elements in the first list represent the number of legs yet to be done and the total profit that far, the rest of the elements represent (in groups of three) the type of plane, the airport where said plane is and the time of the next flight. The second list has size “legs” and each element is either 1 or 0, 1 is said leg is yet to be visited and 0 is already visited. For a given state our code verifies the next possible states, bearing in mind the constraints, meaning that it can only go through legs that belong to that airport and have not been visited yet and verifies if the schedule is possible.

2-The cost function adds to the existing profit the cost of the action selected by the algorithm. As for heuristics our approach was to check for each plane the profit of traveling through every leg yet available and had up the highest value from each one.

3-The A\* algorithm will find the optimal solution as long as the heuristics is admissible, meaning that it doesn’t overestimate the path’s cost from a given node, and consistent, meaning that the estimate is always lesser than or equal to the estimated distance from any neighbor vertex to the estimated goal plus the cost of reaching that neighbor.