Problem Description

Vertices:

Coordinates

# vertices:

Edges:

, connecting and as a straight segment.

Euclidian distance:

# edges:

Trucks:

Crossing an edge :

Cost: . Time: .

Note does not necessarily equal and so does .

Trucks cross the edge uniformly, i.e. its coordinates can be calculated when crossing.

Each truck can take the goods of at most customers.

# trucks:

Drones:

Flying from a point to another (maybe on edges) :

Cost: . Time: .

Each truck carries exactly one drone.

Drones are in charge of taking goods directly to a customer, while trucks do not.

Each drone serves exactly one customer in a take-off.

Truck can stop at any position to send its drone to a customer and wait until it comes back.

Drones always flies straightly.

Customers:

The location of customer is at and the time window is .

All customers must be served.

If the time window is not satisfied with drone arriving at time ,

we have penalty: .

# customers:

Warehouse:

The location is at vertex .

Each truck’s route must start and end at .

Truck can reload as long as it comes back to the warehouse.

Costs:

The truck’s path: .

Drone flies to reach customer , and arrives at .

The cost is:

Formulation

Hyperparameters:

i.e., 8:00-20:00

Edges:

Edges are divided into streets and highways.

For streets:

For highways:

Generally, a vertex can generate several streets but no more than five highways (average: 0.5). However, the warehouse, i.e. vertex 1 can have eight highways. 20% of the vertices have no highways and 12% of customers fall in this group.

Customers:

40% of time windows falls into , i.e. 9:00-11:00 and 14:00-17:00. These customers want to receive their goods in work hours.

40% of the time windows falls into , i.e. 17:00-20:00. These customers want to receive goods when they’re at home.

Algorithms

1. Genetic algorithm
2. Simulated annealing
3. Tabu search