Excercise 4
Implementing a centralized agent

Group №34: Anh-Nghia Khau, Stephane Cayssials

November 8, 2016

1 Solution Representation

1.1 Variables

We have implemented some classes for our solution:

- class **Task_(Task, Action)** : discribe a task to be delivery or pickup.
- class **A(HashMap<Vehicle, LinkedList<Task_>>)**: a map between vehicle and a list of Task_ to be done.
- class **PickupDeliveryProblem(List<Vehicle>, TaskSet)**: where we run the algorithm to create the plan for all agents

1.2 Constraints

Some contraints that we check:

- A task can not be pickup by two different vehicle
- The capacity of the vehicle can not be exceeded
- All tasks have to be delivered at the end.

Since we use a LinkedList to append the task (each time we append the new task into the head), so we do not have to check time. Because we will do all the tasks sequentially from head to the last.

1.3 Objective function

We want to optimize the total cost of all vehicles in the world. For each vehicle we will compute the cost to do all the Task_. Compute the cost of a task is also depend on the nextTask...

```
if (task == PICKUP && nextTask== PICKUP){
 cost += task.pickupCity.distanceTo(nextTask.pickupCity)*v.costPerKm();
} else if (task == DELIVERY && nextTask == PICKUP){
cost += task.deliveryCity.distanceTo(nextTask.pickupCity)*v.costPerKm();
} else if (task == PICKUP && nextTask== DELIVERY){
cost += task.pickupCity.distanceTo(nextTask.deliveryCity)*v.costPerKm();
} else {
cost += task.deliveryCity.distanceTo(nextTask.deliveryCity)*v.costPerKm();
}
```

2 Stochastic optimization

2.1 Initial solution

For the initial solution, we have three approaches:

- SelectInitialSolution_1: Give all tasks to the biggest vehicle
- SelectInitialSolution_2: Give all tasks to all vehicles randomly

2.2 Generating neighbours

For chossing the neighbors, we apply two approaches in pseudo code in the paper. One is **Changing vehicle**: take the first task from the tasks of one vehicle and give it to another vehicle, and the other one is **Changing task order**: change the order of two tasks in the tasks of this vehicle (have to avoid some cases that will be violated the constraint).

2.3 Stochastic optimization algorithm

For each iteration, we will discover some new plan (M plans) from the generating neighbors method. Let's say A is the best one among M plans ("best" based on objective function). Then we will return A with a probability P (a value between 0.3 and 0.5 would be a good choice) or return old A with a probability (1-P). In the case if p = 1, the algorithm converges faster but A maybe a local minimum and it's not very good because we want to find the global minimun.

3 Results

3.1 Experiment 1: Model parameters

Compare between SelectInitialSolution_1 and SelectInitialSolution_2

3.1.1 Setting

England map, 4 vehicles, 30 tasks (all have the same weight 3kg). For the SelectInitialSolution_1, we have to changed the capacity of one vehicle to 90. For the SelectInitialSolution_2, each vehicle has the same capacity (30kg). The algorithm run in a loop 10000 times with P = 0.4.

3.1.2 Observations

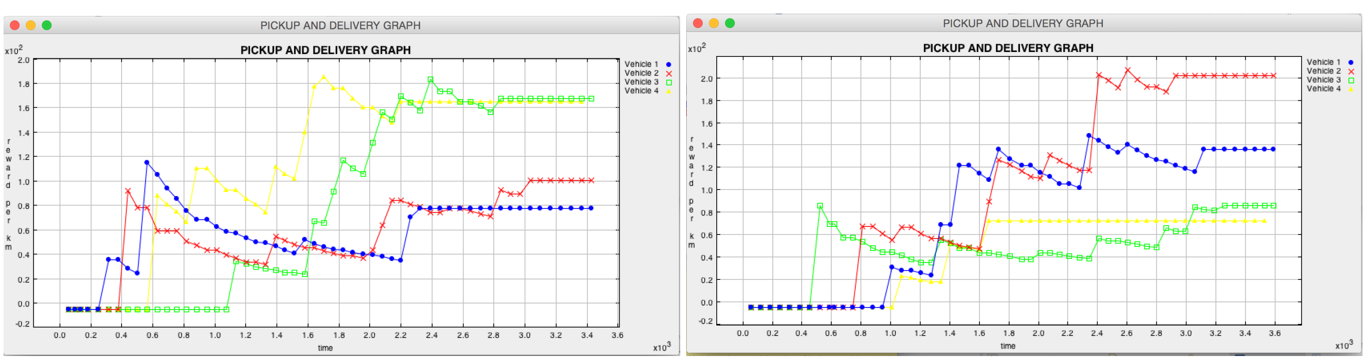

Figure 1: SelectInitialSolution_1(on the left) vs. SelectInitialSolution_2(on the right)

It seems that we have the same result with these two approaches, the SelectInitialSolution_1 has 47k cost and the SelectInitialSolution_2 has 45k cost. There is not much different because the algorithm run 10000 times for looking the best one and in the end, the result gives back the same thing.

3.2 Experiment 2: Different configurations

We want to compare Deliberative and Centralized

3.2.1 Setting

There are 9 tasks, SelectInitialSolution_2 approach for Centralized. The rest is the same as Experiment 1

3.2.2 Observations

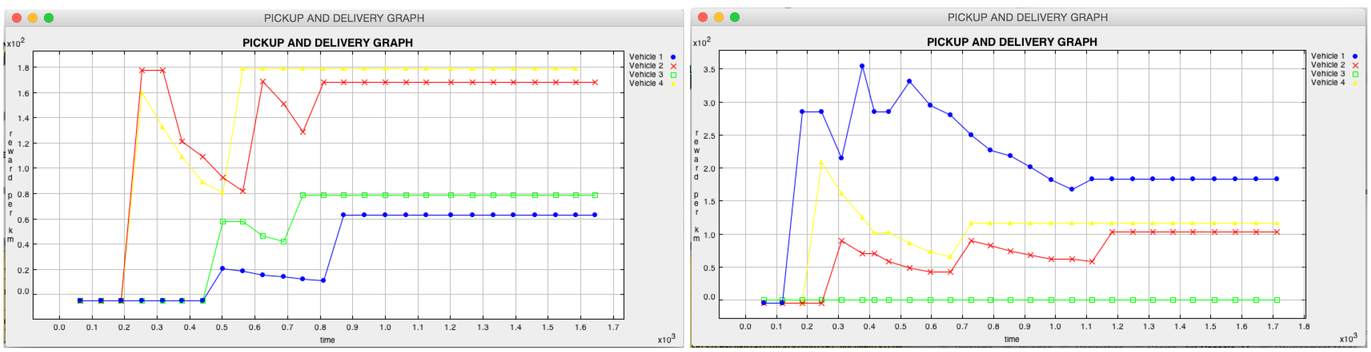

Figure 2: Deliberative (on the left) vs. Centralized(on the right)

We can see that, the centralized has a reward per km bigger than the deliberative in average. Because in this example, we have 4 agents and in deliberative there is always some conflits and they pay the cost each time there was a conflit.

4 Conclusion

The complexity is sure depend on the number of vehicles and the size. Because we have to pass through the list of vehicle and the list of tasks for testing all possible neighbors to find the minimum cost, the complexity is also depend on the number of loop we run the algorithm.

In this problem, we still not have the optimal plan because as I observed, there is a lot of time that an agent pass through a city where on this vehicle there is a task to delivery at this city, but this agent have to carry on the current task and can not deliver the other task. So agent lose the cost for comming back later.