Task Planning for Unmanned Vehicle Delivery

Team 24
ZHU HAOYU, WANG GUANDING, ZENG CONG

1. Why

- With the increasing demand for urban logistics and the maturity of low-speed autonomous driving technologies, Unmanned Vehicle(UV) based delivery has seen increasing practical applications in recent years. Before UVs depart from the warehouse, an efficient scheduling algorithm is required to allocate parcels across UVs and in what sequence they should be delivered, in order to maximize the number of parcels delivered within the limited endurance. In practice, the delivery scale can be very large, and the demand is highly dynamic. Traditional heuristic algorithms often struggle in such large-scale and dynamic settings: solution quality tends to degrade and computational costs grow exponentially. In contrast, a well-trained RL model has the advantage of fast inference. When delivery tasks change, the model can quickly re-plan, meeting the requirements of dynamic environments.
- In this project, we mainly focus on the single-UV delivery problem. For multi-vehicle scenarios in practice, tasks can be clustered and decomposed into several single-vehicle sub-problems. Alternatively, a higher-level agent responsible to allocate tasks or multi-agent reinforcement learning framework can be adopt—an extension we envision as a promising future direction beyond the scope of this project.

2. Conventional Algorithms

- Traditional approaches:
  (a) Exact solvers, such as Integer Linear Programming (ILP) formulations solved via branch-and-bound or branch-and-cut algorithms, guarantee optimal solutions but require massive computational resources and exhibit rapidly increasing solving time as the problem scales.
  (b) Heuristic methods, such as Particle Swarm Optimization (PSO) or Genetic Algorithms(GA), are sensitive to parameter settings and are eazy to being trapped in local optima
- By contrast, RL-based methods concentrate computational cost mainly during the training phase. Once trained, the model can be applied to the problems in the same frame with fast inference. Moreover, RL models often demonstrate a certain degree of generalization: a model trained on smaller-scale problems can often be transferred to larger instances, significantly reducing computation resource consumption. For example, a navigation policy trained in a small grid environment can be transferred and applied to a larger map.

3. Problem Statement

- Map: On a 1×1 map, one depot is randomly generated in each of the four quadrants. Around each depot, 25 delivery points(community express delivery station) are randomly generated following a Gaussian distribution. Each station has 5–15 parcels

to be delivered, uniformly randomly distributed. Map details shown in Fig. 1 and example solution by or-tools shown in Fig.2.

- UV: Endurance is 4, and the number of parcels a UV can carry has no limitation. UV can observe all the information of the map and itself. UV only needs to choose next point to visti in each state, with the goal of delivering as many parcels as possible within the endurance. The UV must return to a depot before exhausting its remaining range.

Note:

1. The UV is not required to deliver all parcels on the map.
2. The UV can return to any depot.
3. In different case, depots and points are fixed, while the number of parcels in each point may vary.

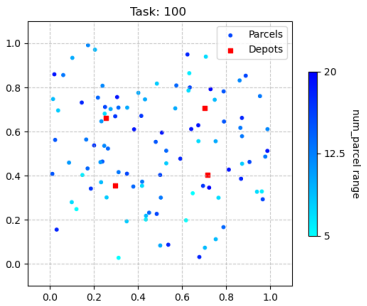

Fig. 1

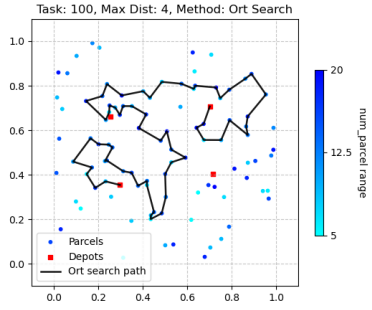

Fig. 2

4. RL Cast

- State (S): The vehicle's current position, remaining range, unvisited points (coordinates, reward values), and visited points (coordinates, reward = 0).
- Action (A): Selecting the next point from the set of unvisited points, or choosing to "return to a depot".
- Transition (T): Updating the vehicle's position and remaining range, marking the current point as visited, and accumulating rewards.
- Reward (R): Gaining reward $r(i) = num\_delivery(i)$ when visiting point i; no additional rewards when choosing to terminate the task.
- End: The vehicle return to a depot.
- Neural Network Structure: A Transformer or Pointer Network is used to encode task and vehicle features, extracting sequence position information. A Fully Connected Layer or MLP is then applied for feature dimension reduction. Invalid actions (already visited points or points exceeding the remaining range) are masked, followed by a softmax layer to output action probabilities.

5. RL Algorithms & Result Evaluation

- RL Training Method: Policy Gradient algorithms such as REINFORCE or PPO.
- Baseline Comparisons: Heuristic algorithms (e.g., PSO) and Google OR-tools.
- Evaluation Metrics: Cumulative reward, solving time, and generalization capability.