Dynamic Vehicle Routing Problem

Project Work III.

Dávid Apagyi
Advisor: Markó Horváth (SZTAKI)

ELTE TTK

9 January, 2025

Introduction

- The Dynamic Vehicle Routing Problem (DVRP) involves adapting vehicle routes in real time to handle new delivery requests (e.g., factory logistics, meal deliveries).
- Seeking a solution that is only statically optimal may not always be the best approach, as it may lack the flexibility needed to handle dynamic changes.
- A more adaptive schedule is often required, one that can efficiently accommodate new requests while minimizing delays.

Introduction

- The Dynamic Vehicle Routing Problem (DVRP) involves adapting vehicle routes in real time to handle new delivery requests (e.g., factory logistics, meal deliveries).
- Seeking a solution that is only statically optimal may not always be the best approach, as it may lack the flexibility needed to handle dynamic changes.
- A more adaptive schedule is often required, one that can efficiently accommodate new requests while minimizing delays.
- During the semester, I focused on a detailed study of a competition problem, and implement a natural algorithm.
- While effective, the algorithm is resource intensive and lacks explicit adaptability to dynamic changes.

Huawei Competition

We follow the notation system from [1]. The input consists of the following:

- A directed graph $G = (F, A)$, where F is the set of factories, and A is the set of arcs connecting the factories. Each arc has a transportation time t_{ij}.
- An order set $O = \{o_i : i = 1, \ldots, N\}$, where each order $o_i = (F_p^i, F_d^i, q^i, t_e^i, t_l^i)$ specifies:
  - F_p^i and F_d^i: the pickup and delivery locations.
  - $q^i = (q^i_{\text{standard}}, q^i_{\text{small}}, q^i_{\text{box}})$: the size of the order in pallets and boxes.
  - t_e^i: the creation time of the order.
  - t_l^i: the committed completion time.
- A fleet of vehicles $V = \{v_k : k = 1, \ldots, K\}$, each with a loading capacity and specific shift times.
- M nodes (factories), where each factory has limited cargo docks and work shifts. Vehicles may need to wait if all docks are busy.

Constraints

The problem must satisfy the following constraints:

1. **Order fulfillment:** All orders must be served.
2. **Completion time:** Orders must be completed before their committed times t_l^i.
3. **Order splitting:** Orders cannot be divided across multiple vehicles unless specified.
4. **Vehicle capacity:** No vehicle can exceed its loading capacity.
5. **Work shifts:** Loading and unloading must occur within shift times. (*We do not take this restriction into account.*)
6. **Dock limitations:** Each factory has limited docks, and vehicles follow a first-come, first-serve rule.

Constraints

The problem must satisfy the following constraints:

1. **Order fulfillment:** All orders must be served.
2. **Completion time:** Orders must be completed before their committed times t_l^i.
3. **Order splitting:** Orders cannot be divided across multiple vehicles unless specified.
4. **Vehicle capacity:** No vehicle can exceed its loading capacity.
5. **Work shifts:** Loading and unloading must occur within shift times. (*We do not take this restriction into account.*)
6. **Dock limitations:** Each factory has limited docks, and vehicles follow a first-come, first-serve rule.

There are also some hidden constraints in the problem that are not explicitly mentioned in the problem statement, but are assumed during the validation process.

Objective function

The problem has two main objectives:

1. Minimize the total delay of orders *(tardiness)*:

$$f_1 = \sum_{i=1}^{N} \max(0, a_i^d - t_i^l),$$

   where a_i^d is the arrival time of order o_i, t_i^l is the committed completion time, and N is the total number of orders.

2. Minimize the average travel distance of vehicles:

$$f_2 = \frac{1}{K} \sum_{k=1}^{K} \sum_{i=1}^{l_k - 1} d_{n_i^k, n_{i+1}^k},$$

   where n_i^k is the i-th node in the route of vehicle v_k, $d_{n_i^k, n_{i+1}^k}$ is the distance between consecutive nodes, and K is the total number of vehicles.

Objective function

The problem has two main objectives:

1. Minimize the total delay of orders *(tardiness)*:

$$f_1 = \sum_{i=1}^{N} \max(0, a_i^d - t_i^l),$$

   where a_i^d is the arrival time of order o_i, t_i^l is the committed completion time, and N is the total number of orders.

2. Minimize the average travel distance of vehicles:

$$f_2 = \frac{1}{K} \sum_{k=1}^{K} \sum_{i=1}^{l_k - 1} d_{n_i^k, n_{i+1}^k},$$

   where n_i^k is the i-th node in the route of vehicle v_k, $d_{n_i^k, n_{i+1}^k}$ is the distance between consecutive nodes, and K is the total number of vehicles.

The overall objective function is: $f = \lambda \cdot f_1 + f_2$, where λ is a large positive constant to prioritize minimizing delays. In the validator, it is fixed as

$$\lambda = \frac{10\,000}{3600}$$

.

Our work in this semester

- Getting familiar with the area, item Understanding the concepts of my advisors general simulation framework. (I rely heavily on it.)
- Incrementally implementing a simple idea: Best Insert.
- This consists of multiple submodules, it is due to technical or development reasons.
- Lot of time spent with debugging due to the many edge cases.
- It already includes some straightforward optimizations, but there are much room to improve.

Results

- We tested the above methods on some smaller input instances.

Results

- We tested the above methods on some smaller input instances.

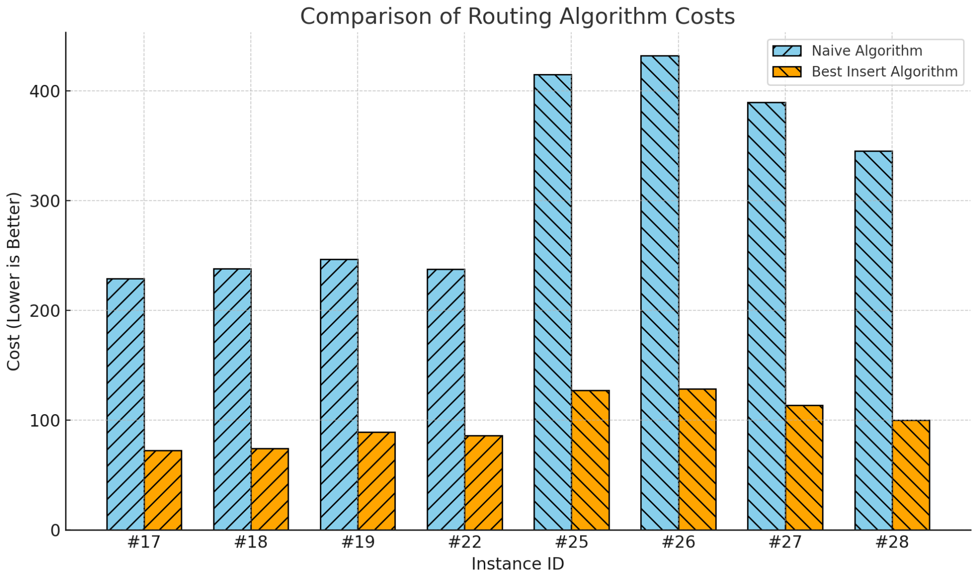

Figure: Comparison of the results between the naive algorithm and the best insert algorithm on smaller instances

Conclusions

- Simple algorithm outperformed naive approach with **68.79% average cost reduction.**
- Smaller instances had minimal vehicle wait times. (Wait times become critical in larger instances, as noted by the advisor.)

Conclusions

- Simple algorithm outperformed naive approach with **68.79% average cost reduction.**
- Smaller instances had minimal vehicle wait times. (Wait times become critical in larger instances, as noted by the advisor.)

For instances that require waiting, further analysis and simulations are needed to explore potential improvements.

Further research

- Implement local search methods to optimize schedules, especially early on with fewer orders.
- Explore modified objective functions (e.g., adding weighted terms) to improve flexibility and regularize solutions.
- Define intuitive policies, such as limiting vehicles per location, that are dynamically adjusted based on requests.
- Developing these ideas further as part of my master's thesis.

Bibliography

Jianye Hao, Jiawen Lu, Xijun Li, Xialiang Tong, Xiang Xiang, Mingxuan Yuan, and Hankz Hankui Zhuo.
Introduction to the dynamic pickup and delivery problem benchmark - ICAPS 2021 competition.
CoRR, abs/2202.01256, 2022.

Markó Horváth and Tímea Tamási.
A general modeling and simulation framework for dynamic vehicle routing, 2024.

Imre Hatala.
Dinamikus jármű útvonaltervezés [dynamic vehicle routing], 2024.
Report written in Hungarian.

Markó Horváth, Tamás Kis, and Péter Györgyi.
A cost function approximation method for dynamic vehicle routing with docking and lifo constraints, 2024.

Ninja Soeffker, Marlin W. Ulmer, and Dirk C. Mattfeld.
Stochastic dynamic vehicle routing in the light of prescriptive analytics: A review.

Thank you for your attention!