# OCAT Duo Assignment

Team Members : 112701006 吳昭泓, 112701035 陳俊翰

---

**Claim**: We have utilized AI to implement the formulation and typeset the content in LaTeX.

## 1. Introduction

Pick-Up and Delivery Problem with Time Window (PDPTW) is a classical extension of Vehicle Routing Problem with Time Window (VRPTW). It considers pickup and delivery pairs, where a total of $|N|$ nodes (including the depot) must be served by a homogeneous fleet of $|K|$ vehicles. While the existing three-index formulation (Section 2) guarantees optimality, its variable complexity $O(|N|^2|K|)$ creates computational bottlenecks for large-scale problems.

To address this issue, Section 3 proposes a new compact two-index formulation that reduces the variable complexity to $O(|N|^2)$. The model adds another series of decision variables, with complexity = $O(|N|)$ which record the index of the first node in the route that visits each node , to explicitly assign vehicles. Finally, Section 4 presents our numerical experiment, supporting the author's efficiency argument.

## 2. Existing Three-Index Formulation

In VRPTW, the two-index decision variables $x_{ij}$ are sufficient. However, in PDPTW, we must track which vehicle visits each node to ensure that each pick-up and its corresponding delivery node are served by the same vehicle, and that the pick-up node is visited first. Therefore, it is essential to incorporate vehicle information into decision variables.

### *2.1. Sets*

- $N = \{0, 1, \ldots, 2n+1\}$ is the set of all nodes, where the nodes 0 and $2n+1$ are the same, denoting the depot.
- $P = \{1, \ldots, n\}$ is the set of all the pick-up nodes.
- $D = \{n+1, \ldots, 2n\}$ is the set of all the delivery nodes.
  Note that the corresponding delivery node of $i \in P$ is assumed to be $n+i$ in the author's formulation.
- $K$ is the set of all vehicles.

### *2.2. Parameters*

- $c_{ij}$ is the travel cost from node $i$ to node $j$, $\forall i \in N, j \in N$.
- $t_{ij}$ is the travel time from node $i$ to node $j$, $\forall i \in N, j \in N$.
- $q_i \begin{cases} = 0, & \text{if } i = 0 \text{ or } 2n+1 \\ > 0, & \text{if } i \in P \\ < 0, & \text{if } i \in D \end{cases}$ is the demand of node $i$, $\forall i \in N$.
  Note that $q_{n+i} = -q_i \ \forall i \in P$.
- $e_i$ is the earliest arrival time at node $i, \forall i \in N$.
- $l_i$ is the latest arrival time at node $i, \forall i \in N$.
- $Cap$ is the vehicle capacity (assume a homogeneous fleet).

### 2.3. *Decision Variables*

- $x_{ijk} = \begin{cases} 1 & \text{if vehicle } k \text{ travels directly from node } i \text{ to node } j, \\ 0 & \text{otherwise.} \end{cases}$
- $B_{ik}$ is the arrival time of vehicle $k$ at node $i$, $\quad \forall i \in N, k \in K$.
- $Q_{ik}$ is load of vehicle $k$ after it visits node $i$, $\quad \forall i \in N, k \in K$.

### 2.4. *Mathematical Model*

$$\min \quad \sum_{k\in K}\sum_{i\in N}\sum_{j\in N} c_{ij}x_{ijk} \tag{1}$$

$$\text{s.t.} \quad \sum_{k\in K}\sum_{j\in N} x_{ijk} = 1, \qquad \forall i \in P \tag{2}$$

$$\sum_{j\in N} x_{ijk} - \sum_{j\in N} x_{n+i,j,k} = 0, \qquad \forall i \in P, k \in K \tag{3}$$

$$\sum_{j\in N} x_{0jk} = 1, \qquad \forall k \in K \tag{4}$$

$$\sum_{i\in N} x_{i,2n+1,k} = 1, \qquad \forall k \in K \tag{5}$$

$$\sum_{j\in N} x_{jik} - \sum_{j\in N} x_{ijk} = 0, \qquad \forall i \in P \cup D, k \in K \tag{6}$$

$$B_{jk} \geq B_{ik} + t_{ij} - M(1 - x_{ijk}), \qquad \forall i \in N, j \in N, k \in K \tag{7}$$

$$Q_{jk} \geq Q_{ik} + q_j - M(1 - x_{ijk}), \qquad \forall i \in N, j \in N, k \in K \tag{8}$$

$$B_{ik} + t_{i,n+i} \leq B_{n+i,k}, \qquad \forall i \in P, k \in K \tag{9}$$

$$e_i \leq B_{ik} \leq l_i, \qquad \forall i \in N, k \in K \tag{10}$$

$$\max(0, q_i) \leq Q_{ik} \leq \min(\text{Cap}, \text{Cap} + q_i), \qquad \forall i \in N, k \in K \tag{11}$$

$$x_{ijk} \in \{0, 1\}, \qquad \forall i \in N, j \in N, k \in K \tag{12}$$

The objective function (1) is set up to minimize the total travel cost. Constraint (2) ensures that each pick-up node is assigned to exactly one vehicle. Constraint (3) ensures that each delivery node is assigned to exactly one vehicle, which is the same as the vehicle of its corresponding pick-up node. Constraints (4) and (5) guarantee that each vehicle leaves and returns to the depot exactly once. Constraint (6) ensures flow conservation. Constraints (7) and (8) confirm the balance of load and time between the nodes. Constraint (9) indicates that each pick-up node must be visited before its corresponding delivery node. Constraints (10) and (11) restrict the capacity and time window. Finally, constraint (12) indicates that $x_{ijk}$ is a binary variable.

## 3. New Two-Index Formulation

In this new formulation, the author removes the vehicle information from existing decision variables; namely, $x_{ijk}$, $B_{ik}$ and $Q_{ik}$ are replaced with $x_{ij}$, $B_i$ and $Q_i$, respectively. Instead, a new series of decision variables $v_i$ is introduced to record which vehicle visits each node.

### 3.1. *Sets*

The definition of the sets is the same as that mentioned in 2.1.

### 3.2. *Parameters*

The definition of the parameters is the same as that mentioned in 2.2.

### 3.3. *Decision Variables*

- $x_{ij} = \begin{cases} 1 & \text{if a vehicle travels directly from node } i \text{ to node } j, \\ 0 & \text{otherwise.} \end{cases}$
- $B_i$ is the arrival time of vehicle at node $i, i \in N$.
- $Q_i$ is the vehicle load after it visits node $i, i \in N$.
- $v_i$ is the identifier of the first node in the route that visits node $i, i \in N \setminus \{0, 2n+1\}$

### 3.4. *Mathematical Model*

$$
\begin{aligned}
\min \quad & \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ij} && && (13)\\
\text{s.t.} \quad & \sum_{j \in N} x_{ij} = 1, && \forall i \in P \cup D && (14)\\
& \sum_{j \in N} x_{ji} = 1, && \forall i \in P \cup D && (15)\\
& B_j \geq B_i + t_{ij} - M(1 - x_{ij}), && \forall i \in N, j \in N && (16)\\
& Q_j \geq Q_i + q_j - M(1 - x_{ij}), && \forall i \in N, j \in N && (17)\\
& B_{n+i} \geq B_i + t_{i,n+i}, && \forall i \in P && (18)\\
& e_i \leq B_i \leq l_i, && \forall i \in N && (19)\\
& \max(0, q_i) \leq Q_i \leq \min(\text{Cap}, \text{Cap} + q_i), && \forall i \in N && (20)\\
& v_{n+i} = v_i, && \forall i \in P && (21)\\
& v_j \geq j \cdot x_{0j}, && \forall j \in P \cup D && (22)\\
& v_j \leq j \cdot x_{0j} - n(x_{0j} - 1), && \forall j \in P \cup D && (23)\\
& v_j \geq v_i + n(x_{ij} - 1), && \forall i, j \in P \cup D && (24)\\
& v_j \leq v_i + n(1 - x_{ij}), && \forall i, j \in P \cup D && (25)\\
& x_{ij} \in \{0, 1\}, && \forall i \in N, j \in N && (26)
\end{aligned}
$$

The object function (13) is to minimize the total travel cost. Constraints (14) and (15) ensure flow conservation. The purposes of constraints (16) to (20) are the same as those of constraints (7) to (11), which are stated in Section 2.4. Constraint (21) ensures that each pick-up and its corresponding delivery node are traveled by the same vehicle. Constraints (22) to (25) enforce the definition of $v_j$—if $j$ is the first node of a vehicle route, $v_j$ must be equal to $j$. Similarly, if a vehicle travels from node $i$ to $j$, $v_i$ should also be equal to $v_j$. Finally, the integrality of $x_{ij}$ is imposed by constraint (26).

## 4. Numerical Experiment

The computational experiments were conducted by Gurobi 12.0.1 in C++ on a Windows 11 laptop with an Intel Core i5 processor (1.30 GHz) and 16 GB of RAM. Since we were unable to locate the data set used in this paper, another widely used set of instances was used instead [1]. In order to match the form of the instances, we modify the formulation as follows.

[1] The author of this paper used the instances similar to those introduced by Ropke, Cordeau, and Laporte (2007), the link of which `http://www.hec.ca/chairedistributique/data` is not currently available. We therefore consider the instances used by Li and Lim (2001) `https://www.sintef.no/projectweb/top/pdptw/li-lim-benchmark/`, which originate from the Solomon (1987) VRPTW instances.

1. Since the total number of vehicles are provided in the dataset, two flow conservation constraints of the depot are added in the new two-index formulation. Constraints (4) and (5) in the existing three-index formulation are modified by the same way.

$$\sum_{j \in N} x_{0j} = V \tag{27}$$

$$\sum_{i \in N} x_{i0} = V \tag{28}$$

, where $V$ is the total number of vehicles.

2. Since the service time at each node is provided in the dataset, constraint (16) are modified to

$$B_j \geq B_i + s_i + t_{ij} - M(1 - x_{ij}), \quad \forall i \in N, j \in N \tag{29}$$

, where $s_i$ is the service time at node i, $\forall i \in N$. Constraint (7) is modified by the same way.

3. Since each pick-up and delivery pair is specified in the dataset, we create an array to store the pair information, rather than assume that the corresponding delivery node of each pick-up node $i$ is $n + i, \forall i \in P$.

4. The big M in constraint (16) was set to $l_i - e_j$ by the author. However, as $B_j \geq e_j$, we hope that $e_j \geq B_i + t_{ij} - M$. Therefore, $M \geq B_i - e_j + t_{ij}$, which is upper bounded by $l_i - e_j + t_{ij}$. The Big M should thus be set to $l_i - e_j + t_{ij}$ so that it is large enough. Moreover, since each node requires a given service time $s_i$, the Big M becomes $l_i - e_j + s_i + t_{ij}$.

5. The big M in constraint (17) was set to $Cap + q_i$. However, we can derive from the constraint that $M \geq Q_i - Q_j + q_j$, which is upper bounded by $Q_i + q_j \leq Cap + q_j$. The Big M should thus be set to $Cap + q_j$ instead.

6. We add several trivial variable fixations in the new two-index formulation to speed up the running time, which are similar as those stated by the author, except for the discrepency stated in 2 and 3, as well as the mislabeling of the depot as $n + 1$ instead of $2n + 1$.

$$x_{ii} = 0, \quad x_{i0} = 0, \quad x_{2n+1,i} = 0, \qquad \forall i \in P \cup D \tag{30}$$

$$x_{i,2n+1} = 0, \quad x_{n+i,0} = 0, \qquad \forall i \in P \tag{31}$$

$$x_{0i} = 0, \qquad \forall i \in D \tag{32}$$

$$x_{ij} = 0 \quad \text{if} \quad e_i + s_i + t_{ij} > l_j, \qquad \forall i, j \in P \cup D \tag{33}$$

$$x_{ij} = 0, \quad x_{i,n+j} = 0, \quad x_{i+n,j+n} = 0 \quad \text{if} \quad q_i + q_j > Cap, \quad \forall i, j \in P \tag{34}$$

Table 1 shows the information for each dataset, including its name, the number of vehicles, the number of customers (half of them are pick-up nodes, and the other half are delivery nodes) and the optimal cost (travel distance).

Table 2 shows the numerical results for each dataset conducted by the existing three-index and the new two-index formulation, including the number of continuous variables (Cont. Vars), the number of binary variables (Binary Vars), and the execution time in seconds. The two formulations both achieve the theoretically optimal solutions, which are provided in the dataset, within the execution time. We have the following observations.

- Among the 9 datasets used, the execution time of the new two-index formulation never exceeds 7 seconds, while the maximum execution time of the existing three-index formulation reaches 1327.57 seconds (approximately 22 minutes). It is evident that the new two-index formulation consistently achieves shorter execution time compared to the existing three-index formulation.

- As the number of vehicles or customers increases, the execution time of the existing three-index formulation grows faster than that of the new two-index formulation. This is mainly due to the higher dimensionality of its variable space and the complexity of associated constraints.
- For those datasets with over 200 customers, the efficiency of the new two-index formulation is 40 to 200 times higher than that of the existing three-index formulation. However, the three-index formulation exhibits the following behaviors.
  - In lc1_2_1, the existing three-index formulation took 82 seconds to obtain the cost = 2696.82635 (infeasible in the primal problem) by dual simplex. In the remaining 83 seconds, it only increased the cost by 8.
  - In lc1_2_5, the existing three-index formulation took 61 seconds to obtain the cost = 2636.75791 (infeasible in the primal problem) by dual simplex. Subsequently, the solution stuck from $61^{\text{th}}$ seconds to $523^{\text{th}}$ seconds.
  - In lc2_2_1, the existing three-index formulation only took 15 seconds to reach the near-optimal cost = 1892.02909.
- When the number of variables increases, regardless of the formulation, the execution time typically increases as well. However, even if the number of vehicles and customers is the same, the execution time varies. This suggests that the number of decision variables is not the only factor that affects execution time.

| **Dataset Information** | | | |
|---|---|---|---|
| **Name** | **Number of Vehicles** | **Number of Customers** | **Optimal Cost** |
| lc101 | 10 | 106 | 828.94 |
| lc105 | 10 | 106 | 828.94 |
| 1c106 | 10 | 106 | 828.94 |
| lc201 | 3 | 106 | 591.56 |
| lc202 | 3 | 106 | 591.56 |
| lc205 | 3 | 106 | 588.88 |
| lc1_2_1 | 20 | 212 | 2704.57 |
| lc1_2_5 | 20 | 214 | 2702.05 |
| lc2_2_1 | 6 | 204 | 1931.44 |

Table 1: Dataset information

| | **Existing Three-Index Formulation** | | | **New Two-Index Formulation** | | |
|---|---|---|---|---|---|---|
| **Data** | **Cont. Vars** | **Binary Vars** | **Time (s)** | **Cont. Vars** | **Binary Vars** | **Time (s)** |
| lc101 | 2160 | 116640 | 7.383 | 324 | 11664 | 0.243 |
| lc105 | 2160 | 116640 | 23.244 | 324 | 11664 | 0.816 |
| 1c106 | 2160 | 116640 | 737.924 | 324 | 11664 | 4.396 |
| lc201 | 624 | 32448 | 0.49 | 312 | 10816 | 0.33 |
| lc202 | 624 | 32448 | 5.038 | 312 | 10816 | 1.198 |
| lc205 | 624 | 32448 | 9.196 | 312 | 10816 | 3.205 |
| lc1_2_1 | 8560 | 915920 | 165.365 | 642 | 45796 | 4.091 |
| lc1_2_5 | 8640 | 933120 | 1327.57 | 648 | 46656 | 6.673 |
| lc2_2_1 | 2472 | 254616 | 222.491 | 618 | 42436 | 4.898 |

Table 2: Comparisoin of execution time between the two formulations